

UNIVERSIDAD NACIONAL DE PIURA
FACULTAD DE CIENCIAS
ESCUELA PROFESIONAL DE INGENIERÍA ELECTRÓNICA Y
TELECOMUNICACIONES



TESIS

**“DISEÑO E IMPLEMENTACION DE UNA LAMPARA DE
FOTOTERAPIA LED CON CONTROL DE LUMINOSIDAD
PARA TRATAMIENTO DE ICTERICIA EN NEONATOS”.**

AUTOR:

Br. JUAN CARLOS CASTRO MONCADA

ASESOR:

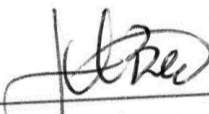
Ing. EDUARDO OMAR AVILA REGALADO

**LÍNEA DE INVESTIGACIÓN: INFORMÁTICA ELECTRÓNICA Y
TELECOMUNICACIONES**


SUB-LÍNEA DE INVESTIGACIÓN: SISTEMA DIGITALES

PIURA JUNIO 2018

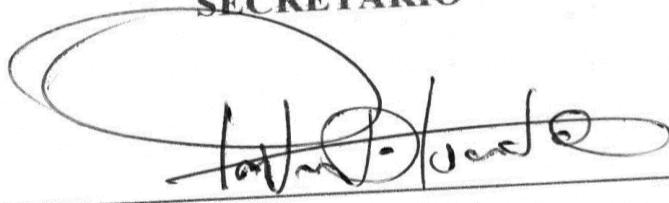
Jurados:



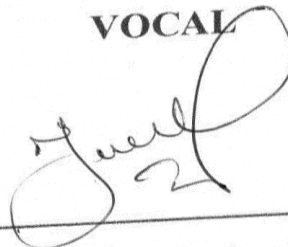
Msc. Franklin Barra Zapata
PRESIDENTE



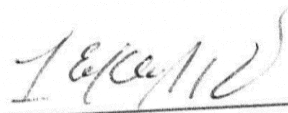
Dr. Carlos Enrique Arellano Ramírez
SECRETARIO



Ing. Miguel Ángel Panduro Alvarado
VOCAL



Br. Juan Carlos Castro Moncada
AUTOR



Ing. Eduardo Omar Ávila Regalado
ASESOR

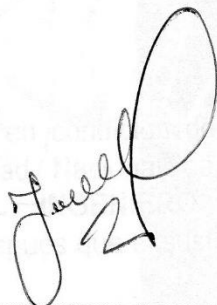
DECLARACION JURADA DE ORIGINALIDAD DE LA TESIS

Yo **Juan Carlos Castro Moncada** Identificado con DNI N°**46203577**, Bachiller de la Escuela Profesional de Ingeniería Electrónica y Telecomunicaciones, de la Facultad de Ciencias y Domiciliado en Calle el Cóndor Mnz. G1 Lt19 de la Provincia de Sullana Departamento de Piura, Celular: 960956798, Email: jccm.0605@Gmail.com

DECLARO BAJO JURAMENTO: que la tesis que presento es original e inédita, no siendo copia parcial ni total de una tesis desarrollada, y/o realizada en el Perú o en el Extranjero, en caso contrario de resultar falsa la información que proporciono, me sujeto a los alcances de lo establecido en el Art, N° 411, del código Penal concordante con el Art, 32° de la Ley N° 27444, y ley del procedimiento Administrativo General y Las Normas Legales de Protección a los derechos de Autor.

En fe de lo cual Firmo la Presente

Piura 12 de Junio del 2018



DNI N°**46203577**



UNIVERSIDAD NACIONAL DE PIURA
FACULTAD DE CIENCIAS



ACTA DE SUSTENTACIÓN 022-2018-D-FC-UNP

Los Miembros del Jurado Calificador que suscriben, reunidos para evaluar la Tesis denominada **"DISEÑO E IMPLEMENTACIÓN DE UNA LAMPARA DE FOTOTERAPIA LED CON CONTROL DE LUMINOSIDAD PARA TRATAMIENTO DE ICTERICIA EN NEONATOS"** presentado por el señor Bachiller **CASTRO MONCADA JUAN CARLOS**, con el asesoramiento del Ing. **Eduardo Omar Ávila Regalado**; oídas las observaciones y respuestas a las preguntas formuladas, y de conformidad al Reglamento de Tesis para obtener el Título Profesional en la Facultad de Ciencias, lo declaran:

APROBADO (x)

DESAPROBADO ()

Con la mención de:

Muy Bueno

(x) En consecuencia, queda en condición de ser ratificado por el Consejo de Facultad de Ciencias de la Universidad Nacional de Piura, y recibir el **TÍTULO PROFESIONAL DE INGENIERO ELECTRÓNICO Y TELECOMUNICACIONES**.

(x) En consecuencia, queda en condición de ser ratificado por el Consejo Universitario de la Universidad Nacional de Piura, y recibir el **TÍTULO PROFESIONAL DE INGENIERO ELECTRÓNICO Y TELECOMUNICACIONES**; después que el sustentante incorpore la sugerencia del Jurado Calificador.

Piura, 19 de mayo de 2018.

[Signature]
MSc. FRANKLIN BARRA ZAPATA
PRESIDENTE DE JURADO DE TESIS

[Signature]
Dr. CARLOS ENRIQUE ARELLANO RAMÍREZ
SECRETARIO DE JURADO DE TESIS

[Signature]
Ing. MIGUEL ÁNGEL PANDURO ALVARADO
VOCAL DE JURADO DE TESIS



Campus Universitario - Urb. Miraflores S/N. Castilla

Esta tesis está dedicada a mis padres, esposa e hijos por el apoyo y comprensión que han tenido conmigo, por darme siempre las fuerzas y el ánimo para salir adelante y culminar mi carrera profesional. Agradezco a los ingenieros que brindaron su apoyo incondicional, para mi formación profesional y asesoramiento de mi tesis...

Juan Carlos Castro Moncada

AGRADECIMIENTOS

Antes que todo quiero agradecer a Dios por darme la fuerza necesaria para poder culminar mi carrera profesional.

A mi familia por apoyarme siempre.

A todos los ingenieros que hicieron de mi un buen profesional y una mejor persona. No puedo dejar de agradecer a los Ingenieros Franklin Barra Zapata, Carlos Enrique Arellano Ramírez, y Miguel Ángel Panduro Alvarado por sus consejos y ayuda desinteresada.

Un agradecimiento especial a mi asesor Ing. Eduardo Omar Ávila Regalado por hacer posible esta tesis.

Gracias a todos los que me han ayudado a realizarme profesionalmente y no hago mención.

Juan Carlos Castro Moncada

INDICE GENERAL

RESUMEN	15
SUMMARY.....	16
CAPÍTULO 1: PLANTEAMIENTO METODOLÓGICO	17
1.1. DESCRIPCIÓN DE LA REALIDAD PROBLEMÁTICA.....	17
1.2. DEFINICIÓN Y LIMITACIONES DEL PROBLEMA	18
1.3. FORMULACIÓN DEL PROBLEMA	18
1.4. OBJETIVOS DE LA INVESTIGACIÓN	18
1.4.1. OBJETIVOS GENERAL.....	18
1.4.2. OBJETIVOS ESPECIFICOS	18
CAPÍTULO 2: MARCO TEÓRICO	19
2.1. ICTERICIA CLÍNICA	19
2.1.1. HIPERBILIRRUBINEMIA NEONATAL	20
2.1.2. TIPOS DE ICTERICIA NEONATAL.....	20
2.1.3. SECUELAS DE LA HIPERBILIRRUBINEMIA	22
2.1.4. METABOLISMO DE LA BILIRRUBINA	23
2.2. FOTOTERAPIA.....	25
2.2.1. HISTORIA	25
2.2.2. METABOLISMO DE LA BILIRRUBINA EN LA FOTOTERAPIA	26
2.2.3. MECANISMO DE LA FOTOTERAPIA	28
2.2.4. PAUTAS PARA FOTOTERAPIA EN NEONATOS	29
2.2.5. FACTORES IMPORTANTES PARA LA EFICACIA DE LA FOTOTERAPIA.....	30
2.3. TECNOLOGÍAS DE ILUMINACIÓN.....	31
2.3.1. GENERALIDADES.....	31
2.3.2. ESTUDIO DEL ÁREA, ESPECTRO Y POTENCIA DE FUENTES LUMÍNICAS	34
2.3.2.1. LÁMPARAS LED.....	34
2.3.2.2. OTROS TIPOS DE ILUMINACIÓN.....	38
2.3.3. SELECCIÓN DE TECNOLOGÍA ADECUADA EN FUNCIÓN DE CALIDAD DE ILUMINACIÓN Y FACILIDAD DE CONTROL	39
2.4. MICROCONTROLADOR	42
2.4.1. CONTROLADOR Y MICROCONTROLADOR	43
2.4.2. DIFERENCIA ENTRE MICROPROCESADOR Y MICROCONTROLADOR.....	44
2.4.3. APLICACIONES DE LOS MICROCONTROLADORES	46
2.4.4. EL MERCADO DE LOS MICROCONTROLADORES	46

2.4.5.	¿QUÉ MICROCONTROLADOR EMPLEAR?	47
2.4.6.	RECURSOS COMUNES A TODOS LOS MICROCONTROLADORES.	50
2.4.6.1.	RECURSOS ESPECIALES.....	54
2.4.7.	HERRAMIENTAS PARA EL DESARROLLO DE APLICACIONES	57
2.5.	BLUETOOTH.....	59
2.5.1.	HISTORIA DE BLUETOOTH.....	59
2.5.2.	PRINCIPIOS DEL BLUETOOTH	59
	La primera empresa que creó un equipo de trabajo para investigar sobre sistemas de comunicación entre dispositivos, fue Ericsson, que allá por el año 1994 comenzó la investigación de una nueva interfaz de bajo consumo y coste, destinada al envío y recepción de datos entre teléfonos móviles y otros dispositivos. [62]	59
2.5.3.	VERSIONES DE BLUETOOTH.....	60
2.5.4.	MÓDULO BLUETOOTH SLAVE (HC-06)	62
2.5.4.1.	DESCRIPCIÓN GENERAL:	62
2.5.4.2.	ESPECIFICACIONES:	63
2.6.	ANTECEDENTES DEL PROBLEMA.....	63
2.7.	ELABORACIÓN DE LA HIPÓTESIS	66
	CAPÍTULO 3: METODOLOGÍA.....	67
3.1.	TIPO DE ESTUDIO.....	67
3.2.	COBERTURA DEL ESTUDIO	68
3.2.1.	POBLACIÓN	68
3.2.2.	MUESTRA.....	68
3.3.	TÉCNICAS E INSTRUMENTOS	68
3.3.1.	TÉCNICAS	68
3.3.2.	INSTRUMENTOS	68
	CAPÍTULO 4:	69
	DISEÑO DE TARJETA ELECTRONICA PARA LAMPARA DE FOTOTERAPIA LED CON CONTROL DE LUMINOSIDAD.....	69
4.1.	DIAGRAMA DE BLOQUES DE LAMPARA LED Y TARJETA ELECTRONICA	69
4.2.	DESCRIPCION DE LAS ETAPAS DEL DISEÑO DE LAMPARA LED DE FOTOTERAPIA.....	71
4.2.1.	ETAPA DE LUMINOSIDAD	71
4.2.1.1.	ÁREA DE ESTUDIO	71
4.2.1.2.	CÁLCULO LUMÍNICO.....	73
4.2.1.3.	DISTRIBUCIÓN LUMÍNICA	78
4.2.1.4.	DISEÑO DE LA MATRIZ DE LEDS PARA LAMPARA DE FOTOTERAPIA.....	80
4.2.2.	DRIVER CONTROLADOR DE POTENCIA DE LOS LEDS	82

4.2.2.1.	TIPOS DE TIRAS DE LEDS	82
4.2.2.2.	DRIVER DE POTENCIA PARA TIRA LED RGB.....	91
4.2.2.3.	DRIVER DE POTENCIA PARA TIRA LED DE LAMPARA PARA FOTOTERAPIA.....	92
4.2.3.	POSICIÓN DE LÁMPARA POR SERVOMOTOR	92
4.2.3.1.	DEFINICIÓN DE SERVOMOTOR.....	92
4.2.3.2.	TIPOS DE SERVOMOTORES	93
4.2.3.3.	FUNCIONAMIENTO DE UN SERVOMOTOR.....	95
4.2.3.4.	CONSUMO DE ENERGÍA.....	98
4.2.3.5.	SERVOMOTOR SELECCIONADO	98
4.2.4.	SENSORES DE DISTANCIA	99
4.2.4.1.	DESCRIPCIÓN DEL FUNCIONAMIENTO DEL SENSOR HC-SR04	99
4.2.5.	PANEL DE CONTROL.....	101
4.2.6.	DISPLAY LCD	102
4.2.7.	MICROCONTROLADOR – ARDUINO UNO	102
4.2.7.1.	ESPECIFICACIONES TECNICAS	103
4.2.7.2.	POWER: PINES DE ALIMENTACION DEL ARDUINO UNO.	104
4.2.7.3.	ENTRADAS Y SALIDAS DIGITALES.....	104
4.2.7.4.	ENTRADAS ANALÓGICAS	105
4.3.	DISEÑO DEL HARDWARE FINAL.....	106
4.4.	DISEÑO DEL SOFTWARE	108
4.5.	SIMULACIONES DEL DISEÑO DE LAMPARA DE FOTOTERAPIA	113
	En las siguientes figuras se visualizaran pantallas de las simulaciones realizadas:	113
4.6.	COSTOS FINALES DEL PROYECTO.....	116
	CAPÍTULO 5: CONCLUSIONES	117
	CAPÍTULO 6: BIBLIOGRAFÍA	118
	ANEXOS.....	124

INDICE DE FIGURAS

Figura 1: Presencia de Ictericia Neonatal [2]	19
Figura 2: Kernícterus o encefalopatía neonatal bilirrubínica [2].....	22
Figura 3: Glóbulos rojos y Hemoglobina [10].....	23
Figura 4: Metabolismo del grupo Hemo [11]	24
Figura 5: Proceso de composición de la bilirrubina [12].....	24
Figura 6: Metabolismo de la bilirrubina directa e indirecta [10].....	24
Figura 7: Metabolismo completo de la Bilirrubina en el organismo [10]	25
Figura 8: Aplicación de rayos del sol para tratamiento de enfermedades [18]	26
Figura 9: Inicios de investigación de la fototerapia científica [18]	26
Figura 10: Cambio de configuración molecular de la bilirrubina mediante fototerapia[15]	27
Figura 11: Metabolismo de la bilirrubina en la fototerapia [16]	27
Figura 12: Espectro de luz y penetración de irradiación de luz en la piel [15]	28
Figura 13: Esquema gráfico de la fototerapia [15]	29
Figura 14: Pautas para fototerapia intensiva en neonatos hospitalizados nacidos con una edad gestacional de 35 semanas o más [15]	30
Figura 15: Aplicación de fototerapia al paciente en toda su superficie corporal [23]	31
Figura 16: Espectro de luz visible [26].....	32
Figura 17: Representación gráfica del estereorradián [28].....	33
Figura 18: Esquema gráfico de la eficiencia luminosa [28]	33
Figura 19: Estructura física del LED [39]	34
Figura 20: Funcionamiento físico del Led [40]	35
Figura 21: Estructura interna del LED y HPLED [33]	35
Figura 22: Distribución espectral relativa: Leds rojos, verdes y azules [33]	36
Figura 23: Reducción del alcance lumínico en el tiempo [43]	37
Figura 24: Relación flujo luminoso con respecto a la temperatura ambiente [46].....	37
Figura 25: Evolución de la iluminación [45]	39
Figura 26: Estructura de un sistema abierto basado en un microprocesador. La disponibilidad de los buses en el exterior permite que se configure a la medida de la aplicación. [55]	45

Figura 27: El microcontrolador es un sistema cerrado. Todas las partes del computador están contenidas en su interior y sólo salen al exterior las líneas que gobiernan los periféricos. [55]	45
Figura 28: Placas de evaluación: (a) LAB-X1, (b) LAB-X2, (c) LAB-X3, todas las placas son ofrecidas por microEngineering Labs, Inc. (http://www.melabs.com)	58
Figura 29: SIMBOLO DE BLUETOOTH. [62]	59
Figura 30: Modulo bluetooth [63]	62
Figura 31: Diagrama de bloques de lámpara para fototerapia led.	69
Figura 32: Diagrama de bloques de tarjeta electrónica para control de lámpara de fototerapia led.	70
Figura 33: Área de estudio.....	72
Figura 34: Relación entre magnitudes radiométricas y fotométricas [53].....	74
Figura 35: Curvas C.I.E. de la eficiencia luminosa espectral [53]	76
Figura 36: Iluminancia del LED	79
Figura 37: Simulación de la distribución rectangular de Leds en MatLab 2010	81
Figura 38: Simulación de niveles de iluminancia (lx) en escala de colores de la distribución	81
Figura 39: Tipos de Led 1 [59]	84
Figura 40: Tipos de Led 2. [59]	84
Figura 41: Cantidad de Led por metro. [59]	84
Figura 42: Consumo Eléctrico de Led. [59]	85
Figura 43: Cantidad de Luz emitente. [59]	86
Figura 44: Tira Led Monocolor. [59]	86
Figura 45: Tira Led RGB. [59]	87
Figura 46: RGB. [59]	87
Figura 47: Tira led RGB. [59]	88
Figura 48: Grado de Protección Tira led RGB. [59]	89
Figura 49: Circuito Internos de tira led Monocolor. [59]	90
Figura 50: Circuito Internos de tira led RGB. [59]	90
Figura 51: Driver de potencia para tira led RGB basado en el MOSFET IRCZ44. [PROPIO]	91
Figura 52: Driver de potencia para tira led para lámpara basado en el MOSFET IRCZ44. [PROPIO]	92
Figura 53: Servomotor. [60]	92

Figura 54: Partes de un Servomotor. [60].....	93
Figura 55: Servomotores Industriales. [60]	93
Figura 56: Servomotores para modelismo y Robótica. [60].....	94
Figura 57: Colores comunes de los cables de un servomotor. [60]	95
Figura 58: Diagrama de bloques del Servomotor. [60]	95
Figura 59: Representación en diagrama de bloques de un sistema de lazo cerrado. [60]	96
Figura 60: Señales de control de un Servomotor. [60]	97
Figura 61: Ejemplos de trenes de pulsos para las posiciones 180°, 90° y 0° en el eje de un servomotor. [60]	97
Figura 62: Servomotor de PROTEUS. [PROPIO]	99
Figura 63: Conexión del servomotor con ARDUINO UNO. [PROPIO]	99
Figura 64: Diagrama de tiempos del sensor hc-sr04. [61].....	101
Figura 65: Diagrama de conexión del sensor hc-sr04 con ARDUINO. [PROPIO]	101
Figura 66: Diagrama de conexión del HC06 y potenciómetros de control de potencia y posición. [PROPIO].....	102
Figura 67: Diagrama de conexión de la pantalla LCD. [PROPIO]	102
Figura 68: ARDUINO UNO con cable de grabación [Propio]	103
Figura 69: ARDUINO UNO - se indica los pines de alimentación [Propio]	104
Figura 70: ARDUINO UNO indicando las entradas y salidas digitales [Propio]	104
Figura 71: ARDUINO UNO indicando Entradas Analógicas [Propio]	105
Figura 72: ARDUINO UNO indicando los pines utilizados para el proyecto [Propio]	106
Figura 73: Diseño de tarjeta electrónica para sistema de control de lámpara de fototerapia led. [Propio].....	107
Figura 74: Diseño de tarjeta electrónica para sistema de control de lámpara de fototerapia led para simulación en PROTEUS. [Propio].....	108
Figura 75: Código para definiciones y declaración de variables [PROPIO]	109
Figura 76: Código para definiciones y declaración de variables [PROPIO]	109
Figura 77: Código de configuraciones iniciales [PROPIO]	109
Figura 78: Código de lazo principal de programa – lectura de puerto serial [PROPIO]	110
Figura 79: Código de modos de trabajo por puerto serial o por potenciómetros [PROPIO]	110
Figura 80: Código para medición de distancia horizontal [PROPIO]	111

Figura 81: Código para medición de distancia vertical 1 [PROPIO]	111
Figura 82: Código para medición de distancia vertical 2 [PROPIO]	111
Figura 83: Código para visualización de datos en pantalla LCD [PROPIO]	112
Figura 84: Mensaje en LCD al Iniciar Programa [PROPIO].....	113
Figura 85: Mensaje en LCD al Iniciar Programa visualiza distancias [PROPIO].....	113
Figura 86: Configurando por Puerto Serial para controlar por potenciómetros [PROPIO]	114
Figura 87: Mensaje en LCD visualizando distancias, Potencias y ángulo del servomotor controlado por potenciómetros. [PROPIO]	114
Figura 88: Mensaje en LCD visualizando distancias, Potencias y ángulo del servomotor controlado por puerto serial USB. [PROPIO]	114
Figura 89: Visualizando las señales PWM para Potencia (150) y ángulo (87°) del servomotor controlado por puerto serial USB. [PROPIO]	115
Figura 90: Visualizando las señales PWM para Potencia (50) y ángulo (95°) del servomotor controlado por puerto serial USB. [PROPIO]	115
Figura 91: Visualizando las señales PWM para Potencia y ángulo del servomotor controlado por potenciómetro [PROPIO]	116

INDICE DE TABLAS

Tabla 1: Causas de la hiperbilirrubinemia neonatal [7].....	21
Tabla 2: Tipos de Ictericia neonatal clasificación y causas [3,8,9]	22
Tabla 3: Pautas para aplicación de tratamiento de fototerapia en recién nacidos a término y pre término [23].....	29
Tabla 4: Tecnologías de Iluminación [25,32].....	38
Tabla 5: Cuadro comparativo de características de funcionamiento y efectividad de varias tecnologías de iluminación [27, 32, 48]	40
Tabla 6: Ventajas y desventajas de las tecnologías de iluminación [48].....	41
Tabla 7: Características de incubadoras más comerciales [49]	71
Tabla 8: Tabla orientativa de las medidas ideales para niños y niñas por edad [50].....	71
Tabla 9: Longitud de onda e Irradiación espectral de equipos de fototerapia [51]	73
Tabla 10: Magnitudes radiométricas y fotométricas y su nomenclatura [52]	74
Tabla 11: Características del led.....	77
Tabla 12: Características del ARDUINO	103
Tabla 13: Tabla de costos finales para realización de un prototipo.....	116

RESUMEN

La fototerapia utiliza la energía lumínica para modificar la forma y la estructura de la bilirrubina, convirtiéndola en moléculas (fotoproductos) que pueden excretarse mediante bilis u orina, por lo que este método terapéutico se ha convertido en el más eficaz para el tratamiento de ictericia en los recién nacidos.

La presente tesis busca diseñar e implementar un PROTOTIPO DE LÁMPARA DE FOTOTERAPIA que tenga incorporado un sistema de control estable, efectivo y fácil de utilizar para los usuarios en el tratamiento de ictericia en neonatos y que pueda ser empleado en el país.

Para la lámpara se selecciona un tira LED RGB la cual se utilizara la luz AZUL, se diseñara el driver de potencia en base a transistores MOSFET, y se controlara mediante señales PWM para variar la corriente y la potencia de los leds. Esta lámpara contara con un control de posición mediante un servomotor que se encargara de variar el ángulo de iluminación de la lámpara.

Para la variación de la potencia de los leds y el ángulo se contara con un panel de control por potenciómetros, por puertos USB y Bluetooth., estas opciones se programa por estos puertos. Por último se cuenta con un LCD para visualizar los datos de distancia horizontal y vertical, potencia y ángulo de la lámpara. Todo esto manejado por el ARDUINO UNO.

En el primer capítulo se describe la problemática y se establece el objetivo general y los objetivos específicos de la tesis, así como la justificación, los alcances y los límites de la misma. El segundo capítulo hace referencia al marco teórico y las definiciones que nos permiten entender desde la base la aplicación desarrollada. El tercer capítulo muestra la metodología para el desarrollo de la investigación.

En el cuarto capítulo se muestra el desarrollo de la aplicación paso a paso tanto en el diseño del hardware como del software, así como simulaciones electrónicas, también se entrega los costos de implementación

El quinto capítulo nos muestra las conclusiones que se llegaron.

PALABRAS CLAVE:

ICTERICIA, RGB, POTENCIA, LUMINOSIDAD, MICROCONTROLADOR, BLUETOOTH, USB, MOSFET, AZUL, PWM, LCD.

SUMMARY

Phototherapy uses light energy to modify the shape and structure of bilirubin, converting it into molecules (photoproducts) that can be excreted through bile or urine, which is why this therapeutic method has become the most effective for the treatment of jaundice in The newborns.

This thesis seeks to design and implement a PHOTOTHERAPY LIGHT PROTOTYPE that incorporates a stable, effective and easy to use control system for users in the treatment of jaundice in neonates and that can be used in the country.

For the lamp, an RGB LED strip is selected, which will use the BLUE light, the power driver will be designed based on MOSFET transistors, and will be controlled by PWM signals to vary the current and the power of the LEDs. This lamp will have a position control through a servomotor that will be responsible for varying the angle of illumination of the lamp.

For the variation of the power of the LEDs and the angle will have a control panel by potentiometers, USB ports and Bluetooth., These options are programmed by these ports. Finally, there is an LCD to visualize the data of horizontal and vertical distance, power and angle of the lamp. All this handled by the ARDUINO ONE.

In the first chapter the problem is described and the general objective and the specific objectives of the thesis are established, as well as the justification, the scope and the limits of it. The second chapter refers to the theoretical framework and the definitions that allow us to understand the application developed from the base. The third chapter shows the methodology for the development of the research.

In the fourth chapter the development of the step-by-step application is shown in both hardware and software design, as well as electronic simulations, implementation costs are also delivered

The fifth chapter shows us the conclusions that were reached.

KEYWORDS:

INK, RGB, POWER, LUMNINOSITY, MICROCONTROLLER, BLUETOOTH, USB, MOSFET, BLUE, PWM, LCD.

CAPÍTULO 1: PLANTEAMIENTO METODOLÓGICO

1.1. DESCRIPCIÓN DE LA REALIDAD PROBLEMÁTICA

La fototerapia utiliza la energía lumínica para modificar la forma y la estructura de la bilirrubina, convirtiéndola en moléculas (fotoproductos) que pueden excretarse mediante bilis u orina, por lo que este método terapéutico se ha convertido en el más eficaz para el tratamiento de ictericia en los recién nacidos. Las lámparas para la fototerapia deben alcanzar un espectro de luz en la región azul cerca de 460 nm, esto es debido a que la penetración de luz en el tejido es mayor en esta región del espectro de luz y aumenta notablemente con una longitud de onda mayor (entre 460 y 490 nm). El objetivo es tener la mayor penetración en el tejido subcutáneo ya que la mayor cantidad de ondas lumínicas absorbidas por la bilirrubina acelera su eliminación [9].

Las lámparas existentes para la fototerapia solo poseen un nivel de potencia lumínica para tratar a los diferentes casos, además que se requiere ciertos accesorios adicionales para tratar una ictericia con niveles de bilirrubina elevados. La distancia entre la fuente lumínica y el paciente también es de mucha importancia ya que de ésta depende la irradiación de luz que llegue al paciente. Las lámparas de fototerapia convencionales (lámparas de tubos fluorescentes o halógenos) han sido las principales tecnologías con las que se ha tratado la ictericia en neonatos, estos tipos de tecnologías tienen la desventaja de producir una gran cantidad de energía calorífica y por lo tanto un gran consumo de energía eléctrica.

Los dispositivos de tecnología Led proporcionan alta irradiación en azul al espectro verde sin generación de calor excesivo, las unidades de diodo electroluminiscentes son eficientes, duraderas y rentables [9].

En la actualidad no existe un sistema de fototerapia que permita regular la posición y potencia lumínica de forma autónoma según sea el caso a tratar.

Aquí la importancia de continuar con la investigación de la tecnología Led que permita desarrollar un tratamiento más efectivo y menos nocivo para los diferentes casos de ictericia en los recién nacidos.

1.2. DEFINICIÓN Y LIMITACIONES DEL PROBLEMA

En este contexto se pretende con este proyecto desarrollar un PROTOTIPO DE LÁMPARA DE FOTOTERAPIA que tenga incorporado un sistema de control estable, efectivo y fácil de utilizar para los usuarios en el tratamiento de ictericia en neonatos y que pueda ser empleado en el país

1.3. FORMULACIÓN DEL PROBLEMA

De acuerdo a lo antes visto podemos formular el problema en forma de pregunta cómo se muestra a continuación:

¿Es posible realizar el diseño e implementación de un prototipo de lámpara de fototerapia led con control de luminosidad para el tratamiento de ictericia en neonatos para nuestro país?

1.4. OBJETIVOS DE LA INVESTIGACIÓN

1.4.1. OBJETIVOS GENERAL

Diseñar implementar un prototipo de lámpara de fototerapia led con control de luminosidad para el tratamiento de ictericia en neonatos.

1.4.2. OBJETIVOS ESPECIFICOS

- Buscar un microcontrolador adecuado que cubra las necesidades para el diseño de la lámpara de fototerapia led.
- Diseñar el programa para el control de luminosidad en LEDs de potencia RGB en un software como BASCOM AVR.
- Diseñar el controlador de corriente para el manejo de los LEDs de potencia desde el microcontrolador.
- Comprobar el funcionamiento del software diseñado mediante la simulación de este.
- Bajar el costo del equipo en comparación con los existentes

CAPÍTULO 2: MARCO TEÓRICO

2.1. ICTERICIA CLÍNICA

La ICTERICIA es la coloración amarillenta de piel, escleras¹ y mucosas (ver figura 1) por aumento de la bilirrubina y su acumulación en los tejidos corporales; se produce debido a que todos los días muere una cierta cantidad de glóbulos rojos en el cuerpo, cuando éstos se descomponen, una sustancia portadora de oxígeno llamada hemoglobina se convierte en bilirrubina. Se aprecia clínicamente cuando la bilirrubina sérica total es superior a 2 mg/dl (34 μ mol/L) en niños o mayor de 5 mg/dl (85 μ mol/L) en neonatos [1].



Figura 1: Presencia de Ictericia Neonatal [2]

“La ictericia nunca es una entidad en sí misma sino el reflejo de una enfermedad específica subyacente con varias etiologías a distintos niveles del metabolismo de la bilirrubina (hiperbilirrubinemia⁴)” [3, 4].

Normalmente, el hígado elimina la bilirrubina de la sangre y la convierte en una forma capaz de ser expulsada del cuerpo al defecar. En los bebés recién nacidos, puede morir un número de glóbulos rojos mayor que en las demás etapas, lo cual genera una mayor cantidad de bilirrubina. Es posible que el hígado del recién nacido sea demasiado inmaduro para manejar bien la eliminación de la bilirrubina, por lo que ésta se acumula en la sangre. Esta acumulación hace que la piel y algunas veces la parte blanca de los ojos se vuelvan amarillentas. Los bebés prematuros tienen hígados especialmente inmaduros, lo cual los hace más propensos a la ictericia [5].

En la mayoría de los casos la ictericia es benigna, pero por su potencial neurotoxicidad, debe ser monitorizada muy de cerca para identificar a los neonatos que puedan

desarrollar hiperbilirrubinemia severa y alteraciones neurológicas inducidas por la bilirrubina [3, 5].

2.1.1. HIPERBILIRRUBINEMIA NEONATAL

La hiperbilirrubinemia es un proceso patológico que consiste en el incremento anormal de la cantidad de bilirrubina en la sangre circulante, lo que puede producir ictericia [6].

La hiperbilirrubinemia neonatal es un problema muy común en recién nacidos.

Cuando no se detecta, se detecta en forma tardía o no se trata de una manera adecuada, puede causar daños permanentes e irreversibles en el cerebro de un recién nacido, causando inclusive la muerte. La concentración de bilirrubina en recién nacido a término normales aumenta desde su nacimiento, llega a su máximo alrededor del tercer día de vida y disminuye hacia valores normales hacia los 7 o 10 días. El neonato normal produce alrededor de 6 a 8 mg/kg/día de bilirrubina (2.5 veces la tasa de producción de bilirrubina del adulto) [1].

Aproximadamente el 50-60% de neonatos presentan hiperbilirrubinemia y la mayoría de los recién nacidos desarrollan ictericia clínica luego del segundo día de vida como expresión de una condición fisiológica [7]. La hiperbilirrubinemia neonatal es muy común debido a los diferentes trastornos que causan esta anomalía, las cuales se describen en la tabla 1.

2.1.2. TIPOS DE ICTERICIA NEONATAL

En la Tabla 1. Se muestra los tipos de Ictericia Neonatal.

CLASIFICACIÓN GENERAL	SOBREPRODUCCIÓN	SUBSECRECIÓN
Hiperbilirrubinemia Trastorno de la maduración (fisiológico)	<ul style="list-style-type: none"> • Circulación enterohepática⁸ exagerada (obstrucción mecánica, peristaltismo reducido, ayuno o subalimentación) • Alimentación materna (enzimas hepáticas inmaduras) 	<ul style="list-style-type: none"> • Galactosemia¹³ • Hipotiroidismo¹⁴ • Tirosinosis¹⁵ • Hipermetioninemia¹⁶ • Obstrucción tóxica (alimentación)
Hiperbilirrubinemia Trastorno hematológico (adquirido)	<ul style="list-style-type: none"> • Incompatibilidad de grupo sanguíneo feto-materna ABO, Rh, otras • Transfusión materno-fetal o feto-fetal⁹ • Galactosemia • Hemólisis inducido por drogas y vitamina K 	
Hiperbilirrubinemia Trastorno hematológico (congénito)	<ul style="list-style-type: none"> • Esferocitosis hereditaria¹⁰ • Sangre extravascular, petequias¹¹, hematoma, hemorragias • Policitemia¹² 	
Hiperbilirrubinemia Trastorno hepático (congénito)		<ul style="list-style-type: none"> • Daño hepato celular secundario¹⁷ a trastornos metabólicos • Disminución de la capacidad hepática (fármacos) • Disminución de la conjugación de bilirrubina (reducción de la actividad de la glucuroniltransferasa¹⁸, inhibidores enzimáticos por drogas y hormonas) • Alteración del transporte de bilirrubina conjugada hacia fuera del hepatocito¹⁹

Tabla 1: Causas de la hiperbilirrubinemia neonatal [7].

TIPOS DE ICTERICIA	CLASIFICACIÓN	CAUSAS
Ictericia por hiperbilirrubinemia no conjugada	- Ictericia fisiológica del recién nacido	- Limitación del hígado para metabolizar el exceso de bilirrubina producida en los primeros días de vida.
	- Ictericia asociada a la lactancia materna	- Incremento de la circulación entero hepática con aumento de la reabsorción de bilirrubina. El diagnóstico es clínico tras la exclusión de otras causas.
	- Ictericias patológicas	
	- Incompatibilidad de factor RH	- Es la causa más frecuente de ictericia neonatal patológica y en el 97% de los casos se debe a isosensibilización ²⁰ para el antígeno Rh D ²¹ . Se ha demostrado que hay paso de glóbulos rojos Rh (+) fetales al torrente sanguíneo materno Rh (-).
	- Incompatibilidad ABO	- Es ocasionada por la interacción entre los anticuerpos maternos anti-A o anti-B de una madre con grupo O, con los eritrocitos A o B del recién nacido. Se estima que el 20% de todos los embarazos se asocian con incompatibilidad ABO, pero la frecuencia de enfermedad hemolítica grave es muy baja.
Ictericia por hiperbilirrubinemia conjugada	- Ictericias por enfermedad celular	
	- Infecciosas	- Se presentan de forma brusca con ictericia y otros síntomas y signos acompañantes: irritabilidad, fiebre, microcefalia, vómitos, etc. Entre ellas están las hepatitis víricas, hepatitis bacterianas, infección del tracto urinario.
	- Ictericia de base metabólica	- Síndrome de colestasis ²² por enfermedades metabólicas como la galactosemia ²³ , tirosinemia ²⁴ , intolerancia a la fructosa, etc., que se acompaña de vómitos, irritabilidad y pobre succión
	- Ictericias por afectación de la vía biliar	- Presentan un fallo en la secreción biliar junto con aumento de la bilirrubina. Suelen presentar un curso no agudo con buen estado general, ictericia, hepatomegalia ²⁵ y heces claras.

Tabla 2: Tipos de Ictericia neonatal clasificación y causas [3,8,9]

2.1.3. SECUELAS DE LA HIPERBILIRRUBINEMIA

KERNICTERUS

Kernicterus, del griego Kern =núcleo e Icterus = amarillo. Constituye la complicación más grave de la ictericia neonatal; es un síndrome neurológico secundario a la acumulación de bilirrubina no conjugada en las células cerebrales (ver figura 2).



Figura 2: Kernícterus o encefalopatía neonatal bilirrubínica [2]

Se denomina “Kernicterus” a la coloración amarilla de los ganglios basales producida por impregnación con bilirrubina, ocurre cuando los niveles de bilirrubina pasan de 20mg/dl en el suero sanguíneo, produciendo alteraciones en las células nerviosas que acaban de sufrir grave deterioro en los días siguientes, además de la coloración amarillenta intensa. Este síndrome rara vez aparece en los niños a término sanos, pero cuanto más inmaduro es el neonato, mayor es la susceptibilidad [8].

El Kernicterus produce secuelas como sordera, crisis convulsivas, alteraciones visuales, alteraciones del lenguaje. Es posible, que la neurotoxicidad de la bilirrubina se manifieste en forma de cuadros más inespecíficos como trastornos del aprendizaje, hiperactividad y otros [7].

2.1.4. METABOLISMO DE LA BILIRRUBINA

La bilirrubina se forma a partir de la hemoglobina (ver figura 3), ésta se encuentra formando parte de los glóbulos rojos (eritrocitos) y su principal función es intervenir en el transporte de oxígeno [1].

La fuente principal de producción de bilirrubina es la hem de la hemoglobina circulante procedente de la hemólisis de los hematíes caducos (80-90% de los casos). La enzima hemoxygenasa oxida el grupo hem y ocasiona la rotura de un anillo de porfirina en su puente inicial alfa-metano, transformándose en globina, hierro, protoporfirina y monóxido de carbono (ver figura 4) [10, 11].

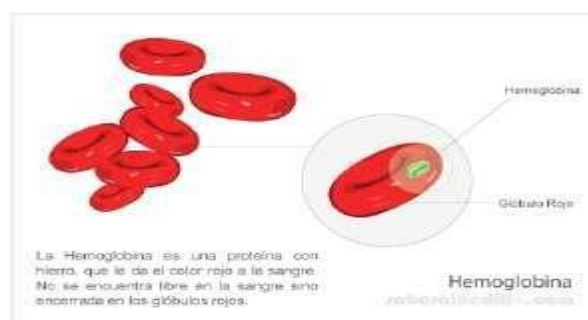


Figura 3: Glóbulos rojos y Hemoglobina [10]

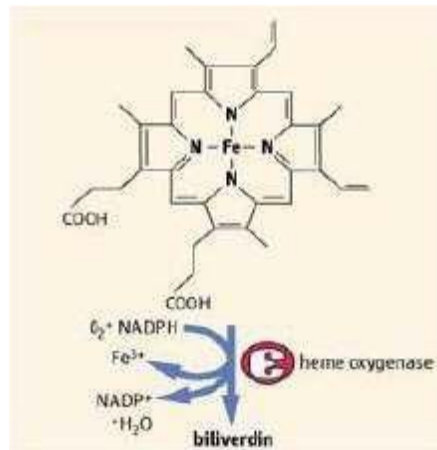


Figura 4: Metabolismo del grupo Hemo [11]

Al destruirse el eritrocito libera la hemoglobina que a través de un proceso metabólico (figura 5) se forman dos tipos de bilirrubina: directa e indirecta. La principal diferencia entre estas dos es que la bilirrubina directa es hidrosoluble (se elimina a través del agua, como la orina y la materia fecal), la bilirrubina indirecta es liposoluble (no se elimina a través del agua y es muy afín a las grasas del cuerpo, particularmente las del cerebro) [10]. Este proceso se explica gráficamente en la figura 6.

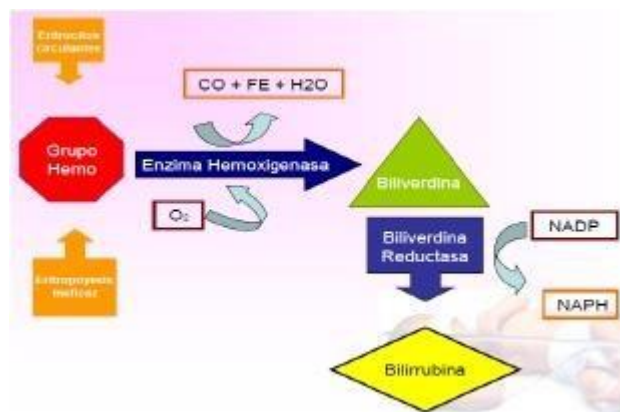


Figura 5: Proceso de composición de la bilirrubina [12]

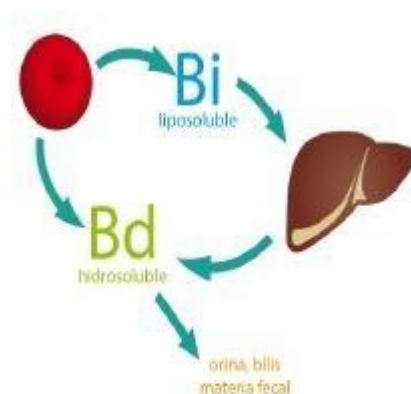


Figura 6: Metabolismo de la bilirrubina directa e indirecta [10]

La bilirrubina indirecta es captada en el hepatocito, luego se conjuga con ácido glucoronico, el cual permite la transformación de esa bilirrubina no conjugada en bilirrubina conjugada, y ésta conjugación es importante porque a través de esta se va a obtener un producto hidrosoluble; este producto entonces va en la bilis [13, 14]. Luego de pasar a la bilis pasa a la flora bacteriana, donde ésta flora vuelve a desconjugar la bilirrubina, y la transforma en otros productos. En una bilirrubina que ya está conjugada se produce urobilinogeno, donde este último es absorbido, y pasa por la sangre y luego al riñón (toda la sangre pasa por el riñón de acuerdo a la circulación); como se indica en la figura 7 [14].

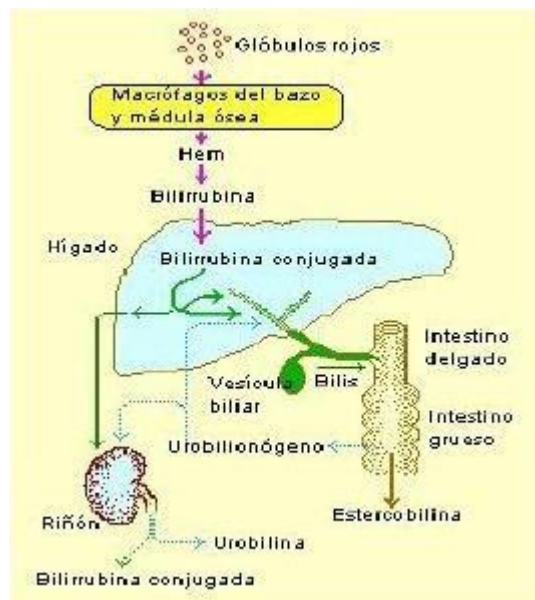


Figura 7: Metabolismo completo de la Bilirrubina en el organismo [10]

2.2. FOTOTERAPIA

2.2.1. HISTORIA

Médicos franceses a mediados del siglo XVIII, pudieron comprobar los efectos beneficiosos del sol en enfermos con infecciones de piel (ver Figura en la figura 8) [17]. Pero la fototerapia científica se inicia con Niels Finser en 1903 quién ganó el Premio Nóbel de Medicina por la aplicación de la UV-terapia en el tratamiento del lupus vulga. Esto marcó el inicio del desarrollo y la investigación de la Fotomedicina (ver figura 9), es decir la aplicación de las radiaciones no ionizantes en medicina [18].



Figura 8: Aplicación de rayos del sol para tratamiento de enfermedades [18]



Figura 9: Inicios de investigación de la fototerapia científica [18]

Hacia finales de la década de los 50 se observó que la exposición de los recién nacidos a la luz solar o a lámparas fluorescentes azules producía una disminución en la concentración sérica de bilirrubina. Desde entonces la fototerapia ha sido un pilar fundamental en el manejo de la hiperbilirrubinemia neonatal [19].

La fototerapia fue evaluada en varios estudios aleatorios realizados entre la década de 1960 y comienzos de la década de 1990 [15]. Es una estrategia terapéutica que ha influenciado notablemente el tratamiento de muchas enfermedades de la piel [20]; y se considera que es la opción terapéutica más económica, segura y con pocas contraindicaciones.

2.2.2 METABOLISMO DE LA BILIRRUBINA EN LA FOTOTERAPIA

La absorción de luz por parte de la forma normal de bilirrubina (4Z, 15Zbilirrubina) genera moléculas de bilirrubina en estado de excitación transitorias.

Estos productos intermedios efímeros pueden reaccionar con el oxígeno para producir productos incoloros de menor peso molecular o pueden reorganizarse y transformarse en isómerosestructurales. El normal 4Z, 15Z-bilirubin se convierte a 4Z, 15E bilirubin (también conocido como photobilirrubina) y a lumirrubina [15, 16], ver figura 10. A diferencia de 4Z, 15Z de la bilirrubina, la photobilirrubina puede ser excretada vía

hepática sin la conjugación, la lumirrubina no es reversible. Aunque la lumirrubina es formada en menos tiempo que la que photobilirrubina, la lumirrubina es eliminada del suero mucho más rápidamente, y es probable que la formación de lumirrubin sea principalmente responsable de la disminución en el suero de la bilirrubina.

Los isómeros de lumirrubina también pueden excretarse en la orina. Los productos de foto oxidación se excretan, principalmente, en orina. Una vez en la bilis, los isómeros configuracionales se revierten espontáneamente a la forma natural 4Z, 15Z de la bilirrubina (proceso ilustrado en la figura 11) [15, 16].

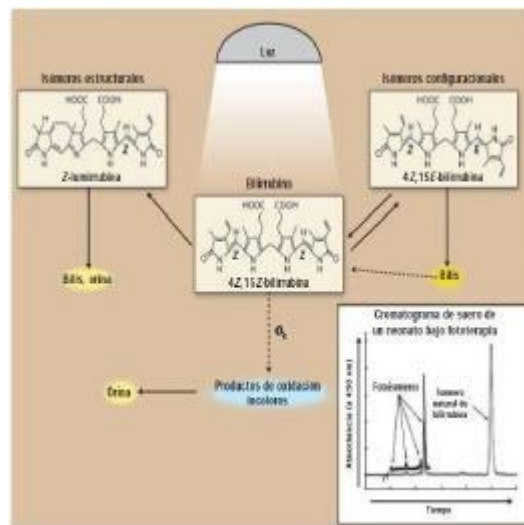


Figura 10: Cambio de configuración molecular de la bilirrubina mediante fototerapia[15]

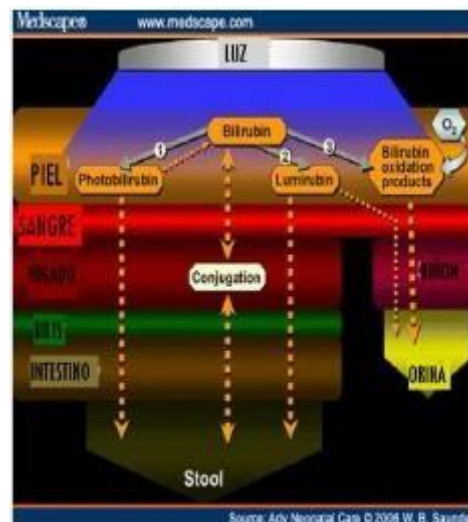


Figura 11: Metabolismo de la bilirrubina en la fototerapia [16]

2.2.3. MECANISMO DE LA FOTOTERAPIA

La fototerapia utiliza la energía lumínica para modificar la forma y la estructura de la bilirrubina, convirtiéndola en moléculas (fotoproductos) que pueden excretarse mediante bilis u orina [15], por lo que este método terapéutico se ha convertido en el más eficaz para el tratamiento de ictericia en los recién nacidos.

Las lámparas para la fototerapia deben alcanzar un espectro de luz en la región azul cerca de 460 nm, esto es debido a que la penetración de luz en el tejido es mayor en esta región del espectro de luz (ver en la figura 12) y aumenta notablemente con una longitud de onda mayor (entre 460 y 490 nm) [15]. El objetivo es tener la mayor penetración en el tejido subcutáneo ya que la mayor cantidad de ondas lumínicas absorbidas por la bilirrubina acelera su eliminación.

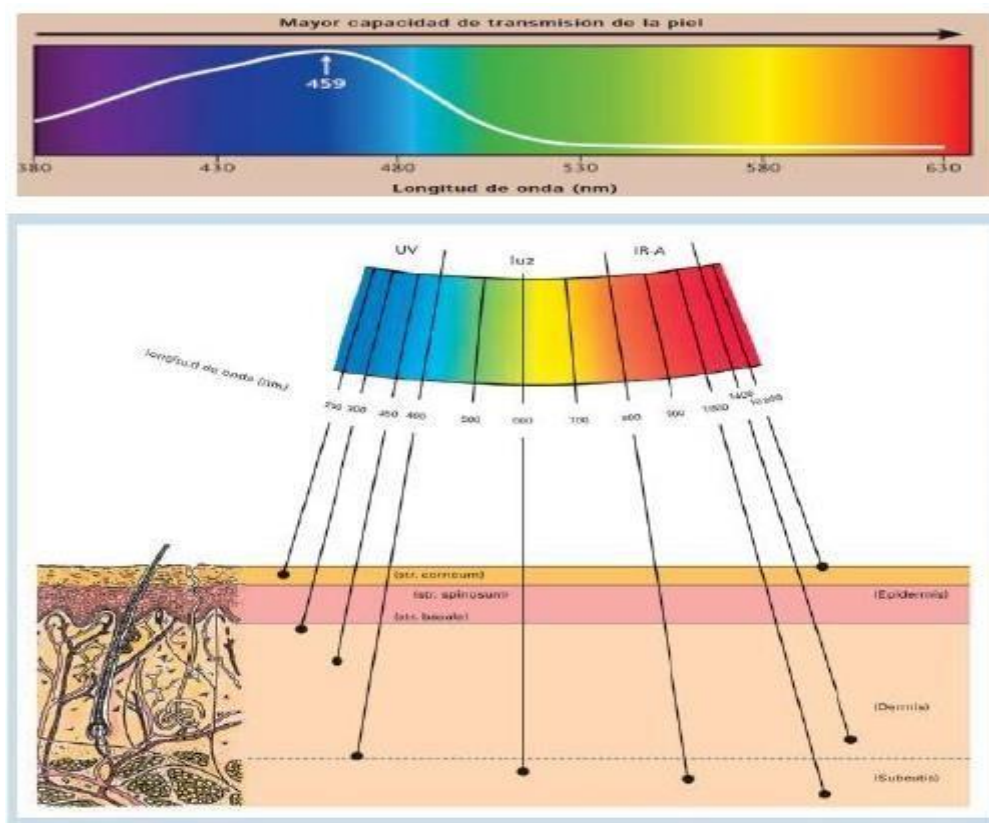


Figura 12: Espectro de luz y penetración de irradiación de luz en la piel [15]

La ictericia en los recién nacidos es diferente en cada individuo por los niveles de bilirrubina que se presenta, es por ello que se debe tratar con diferentes niveles de potencia lumínica según lo indique el tratamiento. La fototerapia es inefectiva para producir reducciones de la concentración de bilirrubinas a niveles menores de 100 mmol/l, por lo tanto no altera esta propiedad [21].

Debe tenerse presente que en la medida que se disminuya la distancia entre el paciente y la fuente, se incrementa el nivel de irradiación y el calor, como se ilustra en la figura 13.



Figura 13: Esquema gráfico de la fototerapia [15]

2.2.4. PAUTAS PARA FOTOTERAPIA EN NEONATOS

Según la Academia Americana de Pediatría, se ofrece una aproximación al tratamiento y se pueden aplicar a los niños de ≥ 35 semanas de edad gestacional.

La aplicación del tratamiento de fototerapia a un neonato se la realiza según el nivel de bilirrubina en la sangre y el peso o la edad a partir de su nacimiento [15].

En los neonatos prematuros, la fototerapia se utiliza con niveles de bilirrubina sérica total mucho menores a las establecidas anteriormente (ver tabla 3). La fototerapia intensiva debe utilizarse cuando el nivel de bilirrubina total se encuentra por encima de la línea de grupo de riesgo indicada para la edad particular [15].

RECIÉN NACIDO PRETERMINO		RECIÉN NACIDO A TÉRMINO	
PESO	FOTOTERAPIA	EDAD	FOTOTERAPIA
< 1000 gr	Profiláctica > 24 h	25 - 48 horas	12 - 15 mg/dl
1000 - 1300 gr	7 - 8 mg/dl	49 - 72 horas	15 - 17 mg/dl
1300 - 1500 gr	8 - 9 mg/dl	>72 horas	>17,5 mg/dl
1500 - 2000 gr	10 - 12 mg/dl		
2000 - 2500 gr	13 - 15 mg/dl		
>2500 gr	15 - 17 mg/dl		

Tabla 3: Pautas para aplicación de tratamiento de fototerapia en recién nacidos a término y pre término [23]

La hiperbilirrubinemia grave (≥ 20 mg/dl) debe ser tratada de inmediato. La agresividad con que esto se lleve a cabo dependerá de las características del niño, de su situación clínica y de los factores de riesgo concomitantes [23]. En las pautas, se recomienda utilizar fototerapia estándar para niveles de bilirrubina sérica total que se encuentren entre 34 a 51 μmol por litro por debajo del intervalo para el cual se recomienda

fototerapia intensiva (ver figura 14), y cuando la bilirrubina sérica total se encuentre por encima de 250 μmol por litro se puede decir que el neonato tiene bajo riesgo de toxicidad bilirrubínica [15].

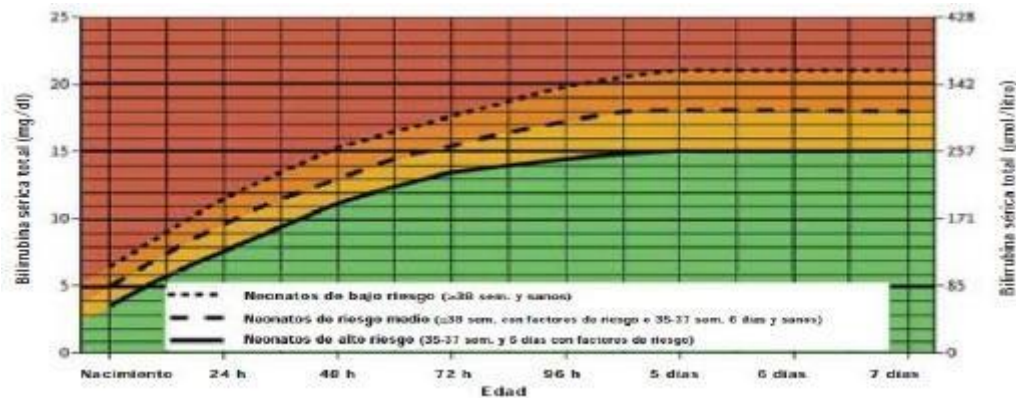


Figura 14: Pautas para fototerapia intensiva en neonatos hospitalizados nacidos con una edad gestacional de 35 semanas o más [15]

2.2.5. FACTORES IMPORTANTES PARA LA EFICACIA DE LA FOTOTERAPIA

La mayor efectividad de la luminoterapia se encuentra en la irradiación con longitud de onda entre 450 y 475 nanómetros administrada por equipos de tubos fluorescentes de luz azul especial, lámparas halógenas, almohadillas con fibra óptica o diodos emisores de luz [2]. La eficacia de la fototerapia está en la dependencia directa de la cantidad de energía liberada en el intervalo de onda correspondiente a la absorción de la luz por la molécula de bilirrubina, del ancho de onda (color) y de la intensidad de la irradiación (energía) de la fuente utilizada, es decir, que la irradiación espectral de la fuente lumínica debe ser mayor a 10 $\text{uW}/\text{cm}^2/\text{nm}$ para tratamiento normal y mayor a 30 $\text{uW}/\text{cm}^2/\text{nm}$ para el tratamiento intensivo [8].

Su efectividad está influenciada por la edad gestacional, peso al nacer, y edad postnatal del niño, como también la etiología de la ictericia y la emisión espectral e intensidad (dosis) de la luz [8], por lo que a continuación se describe maneras para aumentar la eficiencia del tratamiento:

- No se debe interrumpir la lactancia. Se recomienda poner al pecho al recién nacido cada 2 horas para compensar el aumento de la pérdida insensible de agua, que se produce al estar expuesto a la fototerapia.

- Cubrir los Ojos, la luz continua puede tener efectos deletéreos sobre el niño (ver figura 15).
- Control térmico cada 6 horas, ya que es común que a los niños durante el proceso tengan una elevación de temperatura [7].



Figura 15: Aplicación de fototerapia al paciente en toda su superficie corporal [23]

El conocimiento sobre el modo de acción de la luz y sobre el tipo de lámpara utilizada debe ser constantemente monitorizado para su utilización en el tratamiento de la hiperbilirrubinemia neonatal. El aparato que no es monitorizado en lo que se refiere a la irradiación podrá agravar el estado clínico del recién nacido y prolongar, consecuentemente, el tiempo de internación [15,22].

2.3. TECNOLOGÍAS DE ILUMINACIÓN

2.3.1. GENERALIDADES

LUZ

La luz es una manifestación de la energía en forma de radiaciones electromagnéticas capaces de afectar el órgano visual. Las radiaciones visibles se caracterizan por ser capaces de estimular el sentido de la vista y estar comprendidas dentro de una franja de longitud de onda muy estrecha, comprendida aproximadamente entre 380 y 780 nm (ver figura 16) [24, 25].



Figura 16: Espectro de luz visible [26]

Flujo luminoso: El flujo luminoso es la cantidad de luz emitida por una fuente luminosa en la unidad de tiempo (segundo), como se muestra en la figura 18. La unidad de medida del flujo luminoso es el —Lumen (abreviatura lm) y se representa por el símbolo ϕ .

ILUMINACIÓN O ILUMINANCIA: Se define como el flujo luminoso por unidad de superficie, se designa con el símbolo E y se mide en Lux [24]. El Lux es la iluminancia producida por una fuente puntual con intensidad luminosa de una candela sobre una superficie de 1 m² situado a 1 m de la fuente.

$$E = \frac{\phi}{S} = \frac{\text{Flujo luminoso}}{\text{Unidad de superficie}} \text{ LUX} = \frac{\text{LUMEN}}{\text{m}^2} \quad (1)$$

INTENSIDAD LUMINOSA (I): Cantidad de flujo luminoso emitido por cada uno de los rayos que la fuente emite en una determinada dirección por unidad de ángulo sólido. Magnitud que expresa la distribución del flujo luminoso en el espacio.

Esta magnitud se entiende únicamente referida a una determinada dirección y contenida en un ángulo sólido o estéreo que se mide en estereorradianes (ver figura 21). Su unidad de medida es la candela (cd) [28].

ESTERREORRADIÁN: ángulo sólido que corresponde a un casquete esférico cuya superficie es igual al cuadrado del radio de la esfera. A una magnitud de volumen le corresponde un ángulo sólido o estéreo que se mide en estereorradianes (figura 17).

$$I = \frac{\phi}{\omega} = \frac{\text{Flujo luminoso}}{\text{Ángulo sólido}} \text{ cd} = \frac{\text{lumen}}{\text{estereorradián}} \quad (2)$$

EFICIENCIA LUMINOSA: Se define como eficiencia de una fuente luminosa a la relación entre el flujo expresado en lumen, emitido por una fuente luminosa y la potencia absorbida por una lámpara (ver figura 18). Se representa con η y se expresa en Lumen/ Watt [28].

$$\eta = \frac{\varphi}{W} = \frac{\text{lumen}}{\text{Watt}} \quad (3)$$

LUMINANCIA: Relación entre la intensidad luminosa y la superficie aparente vista por el ojo en una dirección determinada. La percepción de la luz es realmente la percepción de diferencias de luminancias [28]. Se usa la letra L para su designación y se mide en cd/m².

$$L = \frac{I}{S_{\text{aparente}}} = \frac{I}{S \cdot \cos \beta} = \frac{\text{cd}}{\text{m}^2} \quad (4)$$

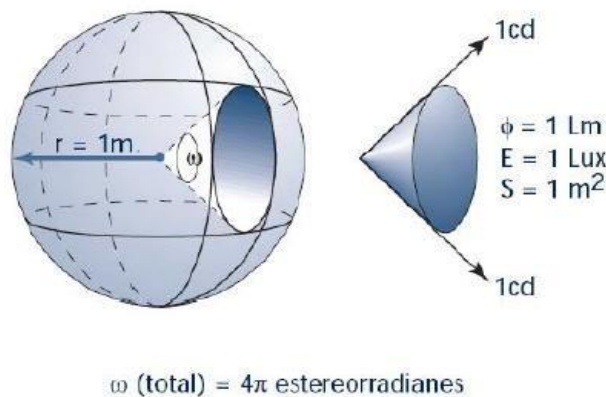


Figura 17: Representación gráfica del estereorradián [28]

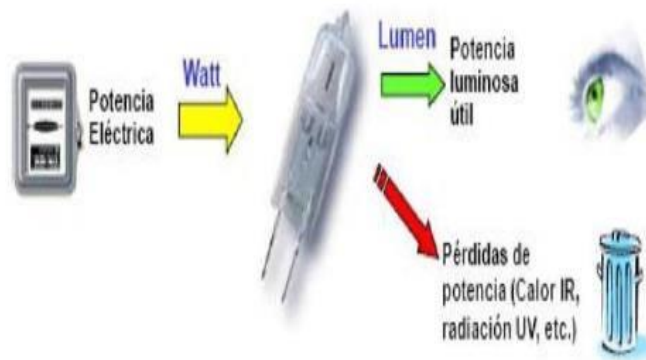


Figura 18: Esquema gráfico de la eficiencia luminosa [28]

HISTORIA DE LA ILUMINACIÓN

El primer gran impacto de la producción de luz artificial lo dieron Edison y Swan en 1879, cuando consiguieron hacer una lámpara útil mediante un filamento alimentado con electricidad. En 1940 se desarrolló la lámpara de descarga fluorescente tubular (el

tubo fluorescente) y posteriormente en 1960 se desarrollaron otro tipo de lámparas de descarga denominadas de halogenuros metálicos [29, 31].

Aunque respecto a los diseños de las lámparas, estos ya se han mejorado mucho, la historia de la iluminación se sigue desarrollando. Ahora el futuro inmediato parece estar en los LEDS (Diodo emisor de luz), que se siguen evolucionando y que están ya presentes en nuestra vida cotidiana [31].

2.3.2. ESTUDIO DEL ÁREA, ESPECTRO Y POTENCIA DE FUENTES LUMÍNICAS

2.3.2.1. LÁMPARAS LED

Un LED es la unión de dos metales semiconductores, uno dopado positivamente y el otro de forma negativa (ver figura 19), tiene las características eléctricas de un diodo, lo cual significa que permite un flujo de corriente en polarización directa y lo impide en polarización inversa [33].

El funcionamiento físico consiste en que, en los materiales semiconductores un electrón al pasar de la banda de conducción a la de valencia, pierde energía; cuando al polarizar directamente un diodo Led conseguimos que por la unión PN sean inyectados huecos en el material tipo N y electrones en el material tipo P; o sea los huecos de la zona p se mueven hacia la zona n y los electrones de la zona n hacia la zona p, produciéndose por consiguiente una inyección de portadores minoritarios, ver figura 20 [41].

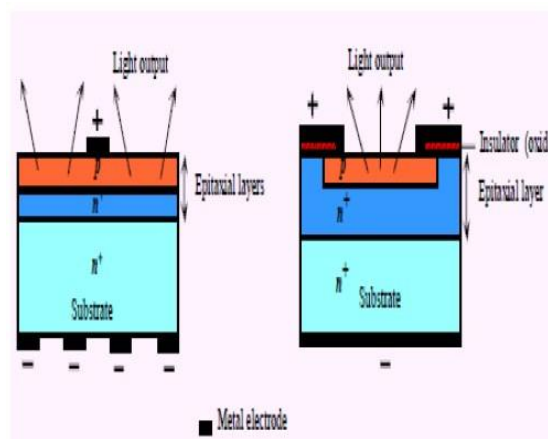


Figura 19: Estructura física del LED [39]

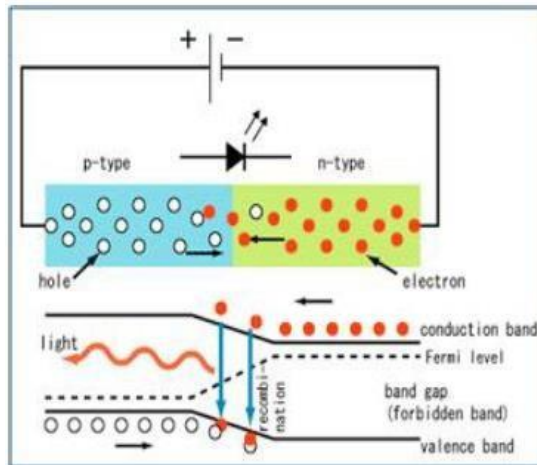


Figura 20: Funcionamiento físico del Led [40]

Los diodos luminiscentes emisores de luz, Leds, se caracterizan por una duración de vida muy larga (una vida media de 50.000 horas), su resistencia a los impactos y un bajo consumo energético. Al ser regulados, el color de luz se mantiene constante. La fuente de luz puntual permite dirigir la luz con toda exactitud. El encapsulado del diodo con material sintético cumple las funciones de protección y de lente como se muestra en la figura 21. La potencia de la radiación del Led disminuye al aumentar la temperatura [33, 38].

Los Leds producen una zona espectral de banda estrecha. La longitud de onda dominante establece el lugar de color del Led y la composición del material semiconductor determina el espectro luminoso entregado como se indica en la figura 22 [33].

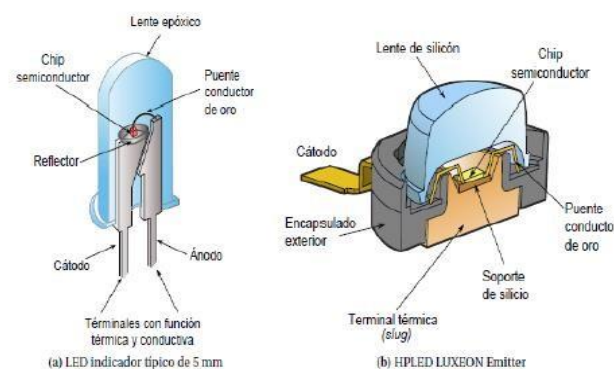


Figura 21: Estructura interna del LED y HPLED [33]

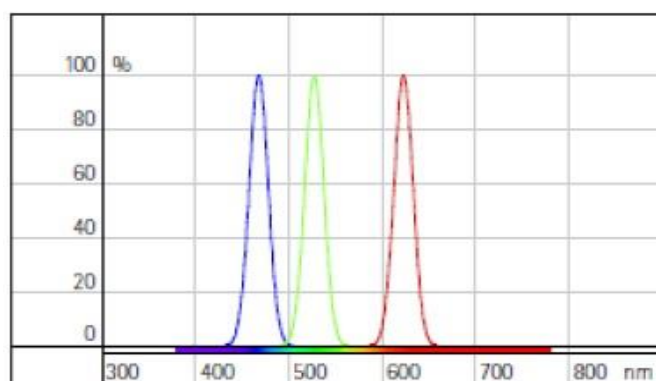


Figura 22: Distribución espectral relativa: Leds rojos, verdes y azules [33]

Actualmente no existe una normalización en el cálculo de vida útil de los Leds.

El Led raramente tiene un fallo total, aunque sí sufre un lento proceso de pérdida en su rendimiento denominado degradación luminosa. Generalmente, un ligero nivel de reducción de luz en un corto espacio de tiempo no es excesivamente importante mientras no llegue al 80% de su valor inicial. La figura 23 muestra la reducción de alcance en tiempo con relación al tanto por ciento de degradación de la intensidad luminosa de la fuente [45].

La relación entre generación y disipación de calor es una limitación clave en la capacidad de producir energía. Los fabricantes especifican una temperatura máxima de trabajo del chip para obtener una vida útil aceptable (ver figura 24).

La resistencia térmica, expresada en $^{\circ}\text{C}/\text{W}$, determina la relación entre la temperatura del Led y la potencia disipada en los contactos o en la superficie [45].

Las últimas tecnologías inciden en el estrecho contacto térmico entre el corazón del Led y la estructura externa. La resistencia térmica se ha reducido alrededor de $10\text{-}15^{\circ}\text{C}/\text{W}$ en los Leds de alta intensidad en comparación con los $100 - 150^{\circ}\text{C}/\text{W}$ para los de baja intensidad [45].

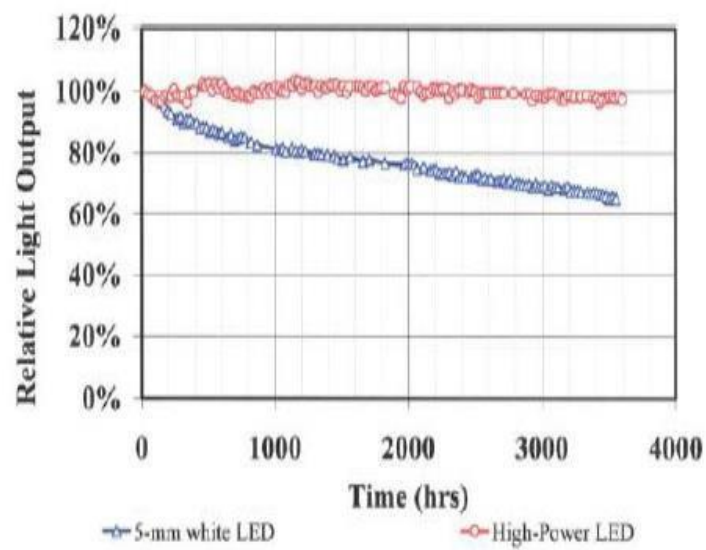


Figura 23: Reducción del alcance lumínico en el tiempo [43]

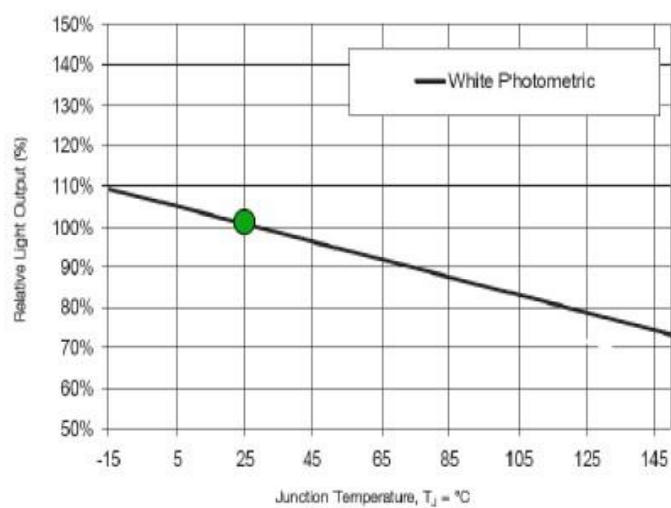


Figura 24: Relación flujo luminoso con respecto a la temperatura ambiente [46]

2.3.2.2. OTROS TIPOS DE ILUMINACIÓN

TIPO DE LÁMPARA	IMAGEN	FUNCIONAMIENTO	CARACTERÍSTICAS
Fluorescente		Las lámparas fluorescentes son fuentes luminosas originadas como consecuencia de una descarga eléctrica en atmósfera de vapor de mercurio a baja presión, en las que la luz se genera por el fenómeno de fluorescencia.	<ul style="list-style-type: none"> - Rendimiento luminoso muy bueno - Vida útil de 10000 horas - Necesita de un dispositivo especial para su funcionamiento - La temperatura de color es de 3200 °K
Incandescente		La incandescencia es un sistema en el que la luz se genera como consecuencia del paso de una corriente eléctrica a través de un filamento conductor. La lámpara estándar (figura 54) es un proyector térmico que contiene un filamento metálico, la aplicación de corriente eléctrica al filamento produce incandescencia	<ul style="list-style-type: none"> - Rendimiento luminoso muy bajo, del orden de 10 a 20 Lm/W. - La temperatura de color es de 2 700 °K. - El rendimiento cromático es del 100 %. - La duración es de unas 1 000 h, excepto las del reflector de vidrio prensado, que es de unas 2 000 h. - No precisan ningún equipo auxiliar para su encendido. - Su utilización se limita al alumbrado general y al localizado de interiores
Halógena		Este grupo utiliza gases halógenos de relleno (normalmente yodo), que disminuyen el fenómeno de evaporización del filamento, ya que el gas halógeno hace que las partículas evaporizadas de wolframio o tungsteno vuelvan a depositarse sobre el filamento de nuevo	<ul style="list-style-type: none"> - Mayor rendimiento luminoso que las incandescentes y llegan a alcanzar los 30 Lm/W. - La temperatura de color es de 3 000 °K. - Pueden alcanzar una vida de 4 000 h. - El flujo luminoso es mayor, debido a que el filamento puede trabajar a mayores temperaturas. Esto es posible gracias a la regeneración del tungsteno. - Debido a sus reducidas dimensiones es posible conseguir un control más preciso del haz luminoso
Sodio de alta presión		Las lámparas de vapor de sodio de alta presión poseen como carga unos gases raros y una amalgama a base de mercurio y sodio, donde la parte correspondiente al gas raro y mercurio se encarga del encendido y de estabilizar la descarga. La lámpara posee un tubo de descarga que se aloja en el interior de una ampolla de vidrio duro, resistente a la intemperie que le sirve de protección y aislamiento eléctrico y térmico.	<ul style="list-style-type: none"> - La eficacia luminosa es francamente buena, del orden de 120 Lm/W. y la temperatura de color resulta ser de 2.200 °K. - La vida media es de unas 15.000 horas, con una depreciación del flujo que no llega al 40%. - La mejora de la reproducción cromática conseguida con estas lámparas, las hace muy apreciadas en alumbrados públicos, y en general en todos aquellos espacios en los que se requiera una iluminación económica sin grandes exigencias cromáticas

Tabla 4: Tecnologías de Iluminación [25,32]

2.3.3. SELECCIÓN DE TECNOLOGÍA ADECUADA EN FUNCIÓN DE CALIDAD DE ILUMINACIÓN Y FACILIDAD DE CONTROL

A través de la historia se han desarrollado nuevas tecnologías de iluminación, cuya tendencia se enfoca cada vez más a una mayor duración (tiempo de vida) y eficiencia (ahorro y mejor uso de la energía eléctrica) de estos medios de iluminación artificial, ver figura 25.

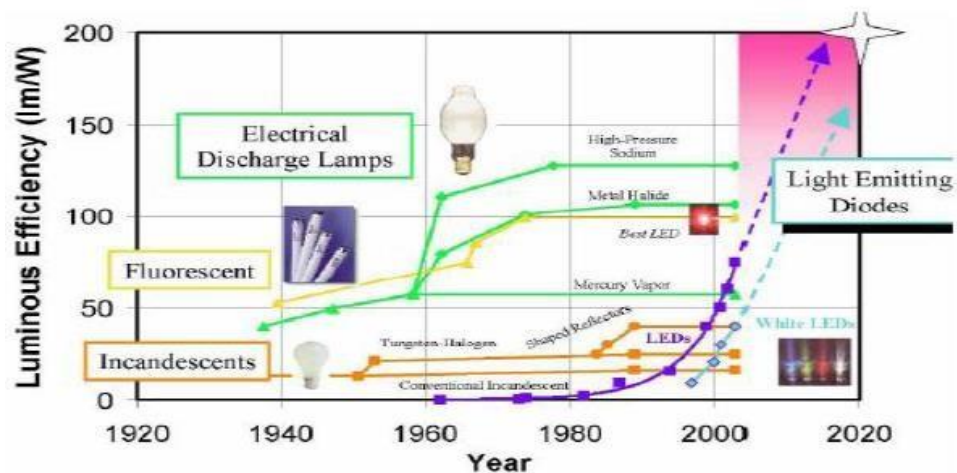



Figura 25: Evolución de la iluminación [45]

Para que el presente proyecto obtenga una mejor eficiencia lumínica y energética, procederemos a realizar una comparativa entre la tecnología Led y los sistemas de iluminación que anteriormente se estudió (tabla 4).

La evolución de la tecnología de iluminación marca una tendencia, producir luz artificial fácil, amigable con el ambiente y duradera. Basado en la comparativa realizada en la tabla 5, podemos decir que los aspectos antes mencionados se agrupan en una sola tecnología.

El Led marca el futuro de la tecnología de iluminación porque es una fuente de luz limpia, es decir, no posee materiales o químicos que contaminen el ambiente. La durabilidad, bajo consumo energético y su gran resistencia a factores externos (golpes, temperatura y luz), hacen que cada día más el Led vaya acaparando el mercado para la iluminación artificial.



Características	Fluorescente	Incandescente	Halógeno	Vapor de sodio	LED
Luz Visible (%)	25	10	15 - 25	30	30
Conducción y Convención (%)	45	20	50	60	70
Emisión Infrarrojo (%)	30	70	15 - 20	20	0
Emisión Ultravioleta (%)	0	0	5 - 20	0,5	0
Eficiencia (lm/W)	60 - 100	15	16 - 25	20-140	> 75
Eficiencia de funcionamiento (%)	80 - 90	100	100	80 - 90	80 - 90
Eficiencia de la Luminaria (%)	50 - 70	30 - 50	30 - 50	40 - 80	80 - 90
Eficiencia total del Sistema (%)	38	4	8	61	58
Vida Útil (horas)	15000	1000	3000	16000	50000
IRC (%)	50 - 95	100	100	60 - 90	> 90

Tabla 5: Cuadro comparativo de características de funcionamiento y efectividad de varias tecnologías de iluminación [27, 32, 48]

El espectro lumínico y la alta capacidad de transformar la energía eléctrica en luz visible hacen que la eficiencia del Led sea bastante buena. Son muchas las ventajas que esta tecnología posee con respecto a las tecnologías antecesoras, pero también tiene algunas desventajas como las que describiremos en la tabla 6.

TECNOLOGÍA DE ILUMINACIÓN	VENTAJAS	DESVENTAJAS
Lámparas Incandescentes	<ul style="list-style-type: none"> • Bajo costo • No necesita de un sistema especial de alimentación (balastro) • Máximo IRC 	<ul style="list-style-type: none"> • Baja eficiencia • Fragilidad a impactos • Vida útil menor a 1000 horas • Emite una gran producción de ondas infrarrojas y ultravioletas
Lámparas Halógenas	<ul style="list-style-type: none"> • Larga vida útil • Alta eficiencia • Debido a sus dimensiones reducidas permite un mayor control del haz de luz 	<ul style="list-style-type: none"> • Fragilidad a impactos • Necesita direccionamiento para enfocar la luz • Se desgasta al encender y apagar • Alta producción de irradiación calorífica
Lámparas Fluorescentes	<ul style="list-style-type: none"> • Larga vida útil • Alta eficiencia energética 	<ul style="list-style-type: none"> • Fragilidad a impactos • Necesita direccionamiento para enfocar la luz • Se desgasta al encender y apagar • Necesita de un sistema especial de alimentación (balastro)
Lámparas de Vapor de Sodio	<ul style="list-style-type: none"> • Larga vida útil • Alta eficiencia • Buena reproducción cromática • Capaz de producir grandes cantidades de lux 	<ul style="list-style-type: none"> • Fragilidad a impactos • Necesita direccionamiento para enfocar la luz • Utiliza químicos contaminantes al medio ambiente • Alta irradiación de ondas caloríficas
Lámparas Led	<ul style="list-style-type: none"> • Muy alta eficiencia y bajo consumo de energía • Mayor vida útil hasta 100 000 horas • Resiste impactos • No emite longitudes de ondas dañinas • No necesita direccionamiento • Puede producir luz blanca y de color 	<ul style="list-style-type: none"> • Costo relativamente elevado • Requiere de un suministro constante de corriente continua para un correcto funcionamiento • Necesita de un sistema especial de alimentación

Tabla 6: Ventajas y desventajas de las tecnologías de iluminación [48]

Obteniendo del estudio las ventajas y desventajas de las más comunes y destacadas tecnologías de iluminación, se puede decir que la tecnología Led supera ampliamente a las demás. Es por esto que para la realización de mi proyecto de tesis escogeré la tecnología Led para el tratamiento de fototerapia en los neonatos con ictericia.

Una de las razones principales que el Led es idóneo para el proyecto es que es la única tecnología que puede alcanzar la emisión de longitud de onda necesaria para la

efectividad del tratamiento de fototerapia. Otra de las razones es que el Led emite luz fría y con esto se elimina la posibilidad de deshidratación masiva en el proceso del tratamiento con fototerapia.

2.4. MICROCONTROLADOR

Muchos de nosotros sabemos qué apariencia tiene una computadora. Usualmente tiene teclado, monitor, CPU (Unidad de Procesamiento Central), impresora y mouse. Este tipo de computadoras, como la Mac o PC, son diseñadas principalmente para comunicarse con humanos. Manejo de base de datos, análisis financieros o incluso procesadores de textos, se encuentran todos dentro de la “gran caja”, que contiene CPU, la memoria, el disco rígido, etc. El verdadero “cómputo”, sin embargo, tiene lugar dentro de la CPU.[54]

Si piensa sobre esto, el único propósito del monitor, teclado, mouse e incluso la impresora, es “conectar” a la CPU con el mundo exterior. ¿Pero usted sabía que hay computadoras alrededor de nosotros, corriendo programas y haciendo cálculos silenciosamente sin interactuar con ningún humano? Estas computadoras están en su auto, en el transbordador espacial, en un juguete, e incluso puede haber uno en su secador de pelo. Llamamos a éstos dispositivos “microcontroladores”. Micro porque son pequeños, y controladores, porque controlan máquinas o incluso otros controladores. Los Microcontroladores, por definición entonces, son diseñados para ser conectados más a máquinas que a personas. Son muy útiles porque usted puede construir una máquina o artefacto, escribir programas para controlarlo, y luego dejarlo trabajar para usted automáticamente. Hay un número infinito de aplicaciones para los microcontroladores. ¡Su imaginación es el único factor limitante! Cientos (sino miles) de variaciones diferentes de microcontroladores están disponibles. Algunos son programados una vez y producidos para aplicaciones específicas, tales como controlar su horno microondas. Otros son “reprogramables”, que quiere decir que pueden ser usados una y varias veces para diferentes aplicaciones (Parallax, 1999). Los microcontroladores son increíblemente versátiles, el mismo dispositivo puede controlar un aeromodelo, una tostadora, o incluso el ABS de su auto (sistema antibloqueo). [55]

2.4.1. CONTROLADOR Y MICROCONTROLADOR

Recibe el nombre de controlador el dispositivo que se emplea para el gobierno de uno o varios procesos. Por ejemplo, el controlador que regula el funcionamiento de un horno dispone de un sensor que mide constantemente su temperatura interna y, cuando traspasa los límites prefijados, genera las señales adecuadas que accionan los efectores que intentan llevar el valor de la temperatura dentro del rango estipulado.

Aunque el concepto de controlador ha permanecido invariable a través del tiempo, su implementación física ha variado frecuentemente. Hace tres décadas, los controladores se construían exclusivamente con componentes de lógica discreta, posteriormente se emplearon los microprocesadores, que se rodeaban con chips de memoria y E/S sobre una tarjeta de circuito impreso. En la actualidad, todos los elementos del controlador se han podido incluir en un chip, el cual recibe el nombre de microcontrolador. Realmente consiste en un sencillo pero completo computador contenido en el corazón (chip) de un circuito integrado. [55]

Un microcontrolador es un circuito integrado de alta escala de integración que incorpora la mayor parte de los elementos que configuran un controlador [55]. Un microcontrolador dispone normalmente de los siguientes componentes:

- Procesador o UCP (Unidad Central de Proceso).
- Memoria RAM para Contener los datos.
- Memoria para el programa tipo ROM/PROM/EPROM.
- Líneas de E/S para comunicarse con el exterior.
- Diversos módulos para el control de periféricos (temporizadores, Puertas Serie y Paralelo, CAD: Conversores Analógico/Digital, CDA: Conversores Digital/Analógico, etc.).
- Generador de impulsos de reloj que sincronizan el funcionamiento de todo el sistema.

Los productos que para su regulación incorporan un microcontrolador disponen de las siguientes ventajas:

- Aumento de prestaciones: un mayor control sobre un determinado elemento representa una mejora considerable en el mismo.
- Aumento de la fiabilidad: al reemplazar el microcontrolador por un elevado número de elementos disminuye el riesgo de averías y se precisan menos ajustes.
- Reducción del tamaño en el producto acabado: La integración del microcontrolador en un chip disminuye el volumen, la mano de obra y los stocks.
- Mayor flexibilidad: las características de control están programadas por lo que su modificación sólo necesita cambios en el programa de instrucciones.

El microcontrolador es en definitiva un circuito integrado que incluye todos los componentes de un computador. Debido a su reducido tamaño es posible montar el controlador en el propio dispositivo al que gobierna. En este caso el controlador recibe el nombre de *controlador empotrado* (embedded controller).[54]

2.4.2. DIFERENCIA ENTRE MICROPROCESADOR Y MICROCONTROLADOR.

El microprocesador es un circuito integrado que contiene la Unidad Central de Proceso (UCP), también llamada procesador, de un computador. La UCP está formada por la Unidad de Control, que interpreta las instrucciones, y el Camino de Datos, que las ejecuta. [55]

Las patitas de un microprocesador sacan al exterior las líneas de sus buses de direcciones, datos y control, para permitir conectarle con la Memoria y los Módulos de E/S y configurar un computador implementado por varios circuitos integrados. Se dice que un microprocesador es un sistema abierto porque su configuración es variable de acuerdo con la aplicación a la que se destine. (Figura 26.)

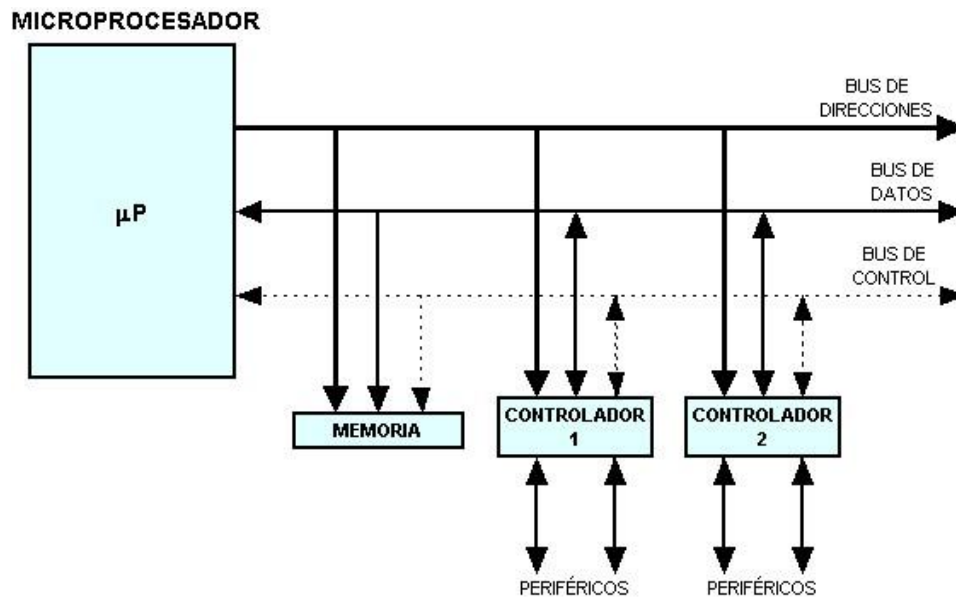


Figura 26: Estructura de un sistema abierto basado en un microprocesador. La disponibilidad de los buses en el exterior permite que se configure a la medida de la aplicación. [55]

Si sólo se dispusiese de un modelo de microcontrolador, éste debería tener muy potenciados todos sus recursos para poderse adaptar a las exigencias de las diferentes aplicaciones. Esta potenciación supondría en muchos casos un despilfarro. En la práctica cada fabricante de microcontroladores oferta un elevado número de modelos diferentes, desde los más sencillos hasta los más poderosos. Es posible seleccionar la capacidad de las memorias, el número de líneas de E/S, la cantidad y potencia de los elementos auxiliares, la velocidad de funcionamiento, etc. Por todo ello, un aspecto muy destacado del diseño es la selección del microcontrolador a utilizar.

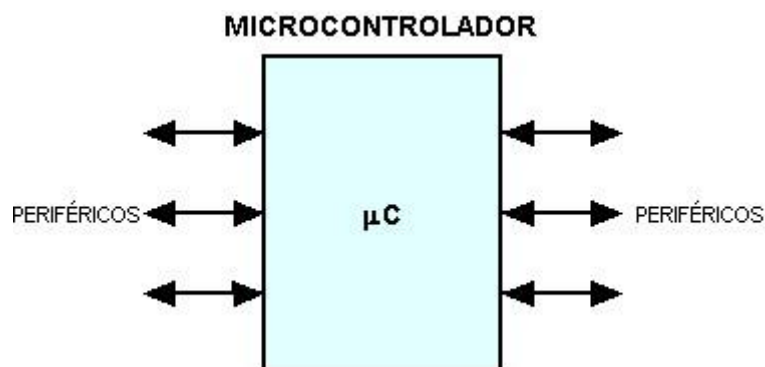


Figura 27: El microcontrolador es un sistema cerrado. Todas las partes del computador están contenidas en su interior y sólo salen al exterior las líneas que gobiernan los periféricos. [55]

2.4.3. APLICACIONES DE LOS MICROCONTROLADORES

Cada vez existen más productos que incorporan un microcontrolador con el fin de aumentar sustancialmente sus prestaciones, reducir su tamaño y coste, mejorar su fiabilidad y disminuir el consumo. Algunos fabricantes de microcontroladores superan el millón de unidades de un modelo determinado producidas en una semana. Este dato puede dar una idea de la masiva utilización de estos componentes.

Los microcontroladores están siendo empleados en multitud de sistemas presentes en nuestra vida diaria, como pueden ser juguetes, horno microondas, frigoríficos, televisores, computadoras, impresoras, módems, el sistema de arranque de nuestro coche, etc. Y otras aplicaciones con las que seguramente no estaremos tan familiarizados como instrumentación electrónica, control de sistemas en una nave espacial, etc. Una aplicación típica podría emplear varios microcontroladores para controlar pequeñas partes del sistema. Estos pequeños controladores podrían comunicarse entre ellos y con un procesador central, probablemente más potente, para compartir la información y coordinar sus acciones, como, de hecho, ocurre ya habitualmente en cualquier PC.

2.4.4. EL MERCADO DE LOS MICROCONTROLADORES

Aunque en el mercado de la microinformática la mayor atención la acaparan los desarrollos de los microprocesadores, lo cierto es que se venden cientos de microcontroladores por cada uno de aquéllos.

Existe una gran diversidad de microcontroladores. Quizá la clasificación más importante sea entre microcontroladores de 4, 8, 16 ó 32 bits. Aunque las prestaciones de los microcontroladores de 16 y 32 bits son superiores a los de 4 y 8 bits, la realidad es que los microcontroladores de 8 bits dominan el mercado y los de 4 bits se resisten a desaparecer. La razón de esta tendencia es que los microcontroladores de 4 y 8 bits son apropiados para la gran mayoría de las aplicaciones, lo que hace absurdo emplear micros más potentes y consecuentemente más caros. Uno de los sectores que más tira del mercado del microcontrolador es el mercado automovilístico. De hecho, algunas de las familias de microcontroladores actuales se desarrollaron pensando en este sector, siendo modificadas posteriormente para adaptarse a sistemas más genéricos. El mercado del automóvil es además uno de los más exigentes: los componentes electrónicos deben operar bajo condiciones extremas de vibraciones, choques, ruido, etc. y seguir siendo

fiables. El fallo de cualquier componente en un automóvil puede ser el origen de un accidente.[54]

En cuanto a las técnicas de fabricación, cabe decir que prácticamente la totalidad de los microcontroladores actuales se fabrican con tecnología CMOS 4 (Complementary Metal Oxide Semiconductor). Esta tecnología supera a las técnicas anteriores por su bajo consumo y alta inmunidad al ruido.

La distribución de las ventas según su aplicación es la siguiente:

Una tercera parte se absorbe en las aplicaciones relacionadas con los computadores y sus periféricos.

La cuarta parte se utiliza en las aplicaciones de consumo (electrodomésticos, juegos, TV, vídeo, etc.)

El 16% de las ventas mundiales se destinó al área de las comunicaciones.

Otro 16% fue empleado en aplicaciones industriales.

El resto de los microcontroladores vendidos en el mundo, aproximadamente un 10% fueron adquiridos por las industrias de automoción.

También los modernos microcontroladores de 32 bits van afianzando sus posiciones en el mercado, siendo las áreas de más interés el procesamiento de imágenes, las comunicaciones, las aplicaciones militares, los procesos industriales y el control de los dispositivos de almacenamiento masivo de datos.[54]

2.4.5. ¿QUÉ MICROCONTROLADOR EMPLEAR?

A la hora de escoger el microcontrolador a emplear en un diseño concreto hay que tener en cuenta multitud de factores, como la documentación y herramientas de desarrollo disponibles y su precio, la cantidad de fabricantes que lo producen y por supuesto las características del microcontrolador (tipo de memoria de programa, número de temporizadores, interrupciones, etc.):

- **Costes.** Como es lógico, los fabricantes de microcontroladores compiten duramente para vender sus productos. Y no les va demasiado mal ya que sin hacer demasiado ruido venden 10 veces más microcontroladores que microprocesadores. Para que nos hagamos una idea, para el fabricante que usa el microcontrolador en su producto una diferencia de precio en el microcontrolador de algunas pesetas es importante (el consumidor deberá pagar además el coste del empaquetado, el de los otros componentes, el diseño del hardware y el

desarrollo del software). Si el fabricante desea reducir costes debe tener en cuenta las herramientas de apoyo con que va a contar: emuladores, simuladores, ensambladores, compiladores, etc. Es habitual que muchos de ellos siempre se decanten por microcontroladores pertenecientes a una única familia.

- **Aplicación.** Antes de seleccionar un microcontrolador es imprescindible analizar los requisitos de la aplicación:
 - **Procesamiento de datos:** puede ser necesario que el microcontrolador realice cálculos críticos en un tiempo limitado. En ese caso debemos asegurarnos de seleccionar un dispositivo suficientemente rápido para ello. Por otro lado, habrá que tener en cuenta la precisión de los datos a manejar: si no es suficiente con un microcontrolador de 8 bits, puede ser necesario acudir a microcontroladores de 16 ó 32 bits, o incluso a hardware de coma flotante. Una alternativa más barata y quizá suficiente es usar librerías para manejar los datos de alta precisión.
 - **Entrada Salida:** para determinar las necesidades de Entrada/Salida del sistema es conveniente dibujar un diagrama de bloques del mismo, de tal forma que sea sencillo identificar la cantidad y tipo de señales a controlar. Una vez realizado este análisis puede ser necesario añadir periféricos hardware externos o cambiar a otro microcontrolador más adecuado a ese sistema.
 - **Consumo:** algunos productos que incorporan microcontroladores están alimentados con baterías y su funcionamiento puede ser tan vital como activar una alarma antirrobo. Lo más conveniente en un caso como éste puede ser que el microcontrolador esté en estado de bajo consumo pero que despierte ante la activación de una señal (una interrupción) y ejecute el programa adecuado para procesarla.
 - **Memoria:** para detectar las necesidades de memoria de nuestra aplicación debemos separarla en memoria volátil (RAM), memoria no volátil (ROM, EPROM, etc.) y memoria no volátil modificable (EEPROM). Este último tipo de memoria puede ser útil para incluir información específica de la aplicación como un número de serie o parámetros de calibración.
 - El tipo de memoria a emplear vendrá determinado por el volumen de ventas previsto del producto: de menor a mayor volumen será conveniente emplear EPROM, OTP y ROM. En cuanto a la cantidad de memoria necesaria puede ser imprescindible realizar una versión preliminar, aunque sea en pseudo-código, de la aplicación y a partir de ella hacer una estimación de cuánta memoria volátil y

no volátil es necesaria y si es conveniente disponer de memoria no volátil modificable.

- **Ancho de palabra:** el criterio de diseño debe ser seleccionar el microcontrolador de menor ancho de palabra que satisfaga los requerimientos de la aplicación. Usar un microcontrolador de 4 bits supondrá una reducción en los costes importante, mientras que uno de 8 bits puede ser el más adecuado si el ancho de los datos es de un byte. Los microcontroladores de 16 y 32 bits, debido a su elevado coste, deben reservarse para aplicaciones que requieran sus altas prestaciones (Entrada/Salida potente o espacio de direccionamiento muy elevado).
- **Diseño de la placa:** la selección de un microcontrolador concreto condicionará el diseño de la placa de circuitos. Debe tenerse en cuenta que quizá usar un microcontrolador barato encarezca el resto de componentes del diseño.

Los microcontroladores más populares se encuentran, sin duda, entre las mejores elecciones:

8048 (Intel). Es el padre de los microcontroladores actuales, el primero de todos. Su precio, disponibilidad y herramientas de desarrollo hacen que todavía sea muy popular.

8051 (Intel y otros). Es sin duda el microcontrolador más popular. Fácil de programar, pero potente. Está bien documentado y posee cientos de variantes e incontables herramientas de desarrollo.

80186, 80188 y 80386 EX (Intel). Versiones en microcontrolador de los populares microprocesadores 8086 y 8088. Su principal ventaja es que permiten aprovechar las herramientas de desarrollo para PC.

68HC11 (Motorola y Toshiba). Es un microcontrolador de 8 bits potente y popular con gran cantidad de variantes.

683xx (Motorola). Surgido a partir de la popular familia 68k, a la que se incorporan algunos periféricos. Son microcontroladores de altísimas prestaciones.

PIC (MicroChip). Familia de microcontroladores que gana popularidad día a día. Fueron los primeros microcontroladores RISC.

Es preciso resaltar en este punto que existen innumerables familias de microcontroladores, cada una de las cuales posee un gran número de variantes.[54]

2.4.6. RECURSOS COMUNES A TODOS LOS MICROCONTROLADORES.

Al estar todos los microcontroladores integrados en un chip, su estructura fundamental y sus características básicas son muy parecidas. Todos deben disponer de los bloques esenciales Procesador, memoria de datos y de instrucciones, líneas de E/S, oscilador de reloj y módulos controladores de periféricos. Sin embargo, cada fabricante intenta enfatizar los recursos más idóneos para las aplicaciones a las que se destinan preferentemente.

En este apartado se hace un recorrido de todos los recursos que se hallan en todos los microcontroladores describiendo las diversas alternativas y opciones que pueden encontrarse según el modelo seleccionado.[54]

ARQUITECTURA BÁSICA

Aunque inicialmente todos los microcontroladores adoptaron la arquitectura clásica de von Neumann, en el momento presente se impone la arquitectura Harvard. La arquitectura de von Neumann se caracteriza por disponer de una sola memoria principal donde se almacenan datos e instrucciones de forma indistinta. A dicha memoria se accede a través de un sistema de buses único (direcciones, datos y control).

La arquitectura Harvard dispone de dos memorias independientes una, que contiene sólo instrucciones y otra, sólo datos. Ambas disponen de sus respectivos sistemas de buses de acceso y es posible realizar operaciones de acceso (lectura o escritura) simultáneamente en ambas memorias. Los microcontroladores PIC responden a la arquitectura Harvard.[54]

EL PROCESADOR O UCP

Es el elemento más importante del microcontrolador y determina sus principales características, tanto a nivel hardware como software.

Se encarga de direccionar la memoria de instrucciones, recibir el código OP de la instrucción en curso, su decodificación y la ejecución de la operación que implica la instrucción, así como la búsqueda de los operandos y el almacenamiento del resultado.

Existen tres orientaciones en cuanto a la arquitectura y funcionalidad de los procesadores actuales.

- **CISC:** Un gran número de procesadores usados en los microcontroladores están basados en la filosofía CISC (Computadores de Juego de Instrucciones Complejo). Disponen de más de 80 instrucciones máquina en su repertorio, algunas de las

cuales son muy sofisticadas y potentes, requiriendo muchos ciclos para su ejecución. Una ventaja de los procesadores CISC es que ofrecen al programador instrucciones complejas que actúan como macros.

- **RISC:** Tanto la industria de los computadores comerciales como la de los microcontroladores están decantándose hacia la filosofía RISC (Computadores de Juego de Instrucciones Reducido). En estos procesadores el repertorio de instrucciones máquina es muy reducido y las instrucciones son simples y, generalmente, se ejecutan en un ciclo.
- La sencillez y rapidez de las instrucciones permiten optimizar el hardware y el software del procesador.
- **SISC:** En los microcontroladores destinados a aplicaciones muy concretas, el juego de instrucciones, además de ser reducido, es “específico”, o sea, las instrucciones se adaptan a las necesidades de la aplicación prevista. Esta filosofía se ha bautizado con el nombre de SISC (Computadores de Juego de Instrucciones Específico).

• **Memoria**

En los microcontroladores la memoria de instrucciones y datos está integrada en el propio chip. Una parte debe ser no volátil, tipo ROM, y se destina a contener el programa de instrucciones que gobierna la aplicación. Otra parte de memoria será tipo RAM, volátil, y se destina a guardar las variables y los datos.

Hay dos peculiaridades que diferencian a los microcontroladores de los computadores personales:

- No existen sistemas de almacenamiento masivo como disco duro o disquetes.
- Como el microcontrolador sólo se destina a una tarea en la memoria ROM, sólo hay que almacenar un único programa de trabajo.

La RAM en estos dispositivos es de poca capacidad pues sólo debe contener las variables y los cambios de información que se produzcan en el transcurso del programa. Por otra parte, como sólo existe un programa activo, no se requiere guardar una copia del mismo en la RAM pues se ejecuta directamente desde la ROM.

Los usuarios de computadores personales están habituados a manejar Megabytes de memoria, pero, los diseñadores con microcontroladores trabajan con capacidades de

ROM comprendidas entre 512 bytes y 8 k bytes y de RAM comprendidas entre 20 y 512 bytes.

Según el tipo de memoria ROM que dispongan los microcontroladores, la aplicación y utilización de los mismos es diferente. Se describen las cinco versiones de memoria no volátil que se pueden encontrar en los microcontroladores del mercado.

- **ROM con máscara**

Es una memoria no volátil de sólo lectura cuyo contenido se graba durante la fabricación del chip. El elevado coste del diseño de la máscara sólo hace aconsejable el empleo de los microcontroladores con este tipo de memoria cuando se precisan cantidades superiores a varios miles de unidades.

- **OTP**

El microcontrolador contiene una memoria no volátil de sólo lectura “programable una sola vez” por el usuario. OTP (One Time Programmable). Es el usuario quien puede escribir el programa en el chip mediante un sencillo grabador controlado por un programa desde un PC. La versión OTP es recomendable cuando es muy corto el ciclo de diseño del producto, o bien, en la construcción de prototipos y series muy pequeñas. Tanto en este tipo de memoria como en la EPROM, se suele usar la encriptación mediante fusibles para proteger el código contenido.

- **EPROM**

Los microcontroladores que disponen de memoria EPROM (Erasable Programmable Read OnIy Memory) pueden borrarse y grabarse muchas veces. La grabación se realiza, como en el caso de los OTP, con un grabador gobernado desde un PC. Si, posteriormente, se desea borrar el contenido, disponen de una ventana de cristal en su superficie por la que se somete a la EPROM a rayos ultravioleta durante varios minutos. Las cápsulas son de material cerámico y son más caros que los microcontroladores con memoria OTP que están hechos con material plástico.

- **EEPROM**

Se trata de memorias de sólo lectura, programables y borrables eléctricamente EEPROM (Electrical Erasable Programmable Read OnIy Memory). Tanto la programación como el borrado, se realizan eléctricamente desde el propio grabador y bajo el control programado de un PC. Es muy cómoda y rápida la operación de grabado y la de borrado. No disponen de ventana de cristal en la superficie.

Los microcontroladores dotados de memoria EEPROM una vez instalados en el circuito, pueden grabarse y borrarse cuantas veces se quiera sin ser retirados de dicho

circuito. Para ello se usan “grabadores en circuito” que confieren una gran flexibilidad y rapidez a la hora de realizar modificaciones en el programa de trabajo.

El número de veces que puede grabarse y borrarse una memoria EEPROM es finito, por lo que no es recomendable una reprogramación continua. Son muy idóneos para la enseñanza y la Ingeniería de diseño.

Se va extendiendo en los fabricantes la tendencia de incluir una pequeña zona de memoria EEPROM en los circuitos programables para guardar y modificar cómodamente una serie de parámetros que adecuan el dispositivo a las condiciones del entorno. Este tipo de memoria es relativamente lenta.

○ **FLASH**

Se trata de una memoria no volátil, de bajo consumo, que se puede escribir y borrar. Funciona como una ROM y una RAM pero consume menos y es más pequeña. A diferencia de la ROM, la memoria FLASH es programable en el circuito. Es más rápida y de mayor densidad que la EEPROM.

La alternativa FLASH está recomendada frente a la EEPROM cuando se precisa gran cantidad de memoria de programa no volátil. Es más veloz y tolera más ciclos de escritura/borrado.

Las memorias EEPROM y FLASH son muy útiles al permitir que los microcontroladores que las incorporan puedan ser reprogramados “en circuito”, es decir, sin tener que sacar el circuito integrado de la tarjeta. Así, un dispositivo con este tipo de memoria incorporado al control del motor de un automóvil permite que pueda modificarse el programa durante la rutina de mantenimiento periódico, compensando los desgastes y otros factores tales como la compresión, la instalación de nuevas piezas, etc. La reprogramación del microcontrolador puede convertirse en una labor rutinaria dentro de la puesta a punto.

● **Puertas de Entrada y Salida**

La principal utilidad de las patitas que posee la cápsula que contiene un microcontrolador es soportar las líneas de E/S que comunican al computador interno con los periféricos exteriores.

Según los controladores de periféricos que posea cada modelo de microcontrolador, las líneas de E/S se destinan a proporcionar el soporte a las señales de entrada, salida y control.

- **Reloj principal**

Todos los microcontroladores disponen de un circuito oscilador que genera una onda cuadrada de alta frecuencia, que configura los impulsos de reloj usados en la sincronización de todas las operaciones del sistema.

Generalmente, el circuito de reloj está incorporado en el microcontrolador y sólo se necesitan unos pocos componentes exteriores para seleccionar y estabilizar la frecuencia de trabajo. Dichos componentes suelen consistir en un cristal de cuarzo junto a elementos pasivos o bien un resonador cerámico o una red R-C.

Aumentar la frecuencia de reloj supone disminuir el tiempo en que se ejecutan las instrucciones pero lleva aparejado un incremento del consumo de energía.[54]

2.4.6.1. RECURSOS ESPECIALES

Cada fabricante oferta numerosas versiones de una arquitectura básica de microcontrolador. En algunas amplía las capacidades de las memorias, en otras incorpora nuevos recursos, en otras reduce las prestaciones al mínimo para aplicaciones muy simples, etc. La labor del diseñador es encontrar el modelo mínimo que satisfaga todos los requerimientos de su aplicación. De esta forma, minimizará el coste, el hardware y el software.

Los principales recursos específicos que incorporan los microcontroladores son:

Temporizadores o “Timers”.

- Perro guardián o “Watchdog”.
- Protección ante fallo de alimentación o “Brownout”.
- Estado de reposo o de bajo consumo.
- Conversor A/D.
- Conversor D/A.
- Comparador analógico.
- Modulador de anchura de impulsos o PWM.
- Puertas de E/S digitales.
- Puertas de comunicación.

A continuación se definen cada uno de los recursos antes mencionados:

- **Temporizadores o “Timers”**

Se emplean para controlar periodos de tiempo (temporizadores) y para llevar la cuenta de acontecimientos que suceden en el exterior (contadores).

Para la medida de tiempos se carga un registro con el valor adecuado y a continuación dicho valor se va incrementando o decrementando al ritmo de los impulsos de reloj o algún múltiplo hasta que se desborde y llegue a 0, momento en el que se produce un aviso.

Cuando se desean contar acontecimientos que se materializan por cambios de nivel o flancos en alguna de las patitas del microcontrolador, el mencionado registro se va incrementando o decrementando al ritmo de dichos impulsos.

- **Perro guardián o “Watchdog”**

Cuando el computador personal se bloquea por un fallo del software u otra causa, se pulsa el botón del reset y se reinicializa el sistema. Pero un microcontrolador funciona sin el control de un supervisor y de forma continuada las 24 horas del día. El Perro guardián consiste en un temporizador que, cuando se desborda y pasa por 0, provoca un reset automáticamente en el sistema.

Se debe diseñar el programa de trabajo que controla la tarea de forma que refresque o inicialice al Perro guardián antes de que provoque el reset. Si falla el programa o se bloquea, no se refrescará al Perro guardián y, al completar su temporización, “ladrará y ladrará” hasta provocar el reset.

- **Protección ante fallo de alimentación o “Brownout”**

Se trata de un circuito que resetea al microcontrolador cuando el voltaje de alimentación (VDD) es inferior a un voltaje mínimo (“brownout”). Mientras el voltaje de alimentación sea inferior al de brownout el dispositivo se mantiene reseteado, comenzando a funcionar normalmente cuando sobrepasa dicho valor.

- **Estado de reposo ó de bajo consumo**

Son abundantes las situaciones reales de trabajo en que el microcontrolador debe esperar, sin hacer nada, a que se produzca algún acontecimiento externo que le ponga de nuevo en funcionamiento. Para ahorrar energía, (factor clave en los aparatos portátiles), los microcontroladores disponen de una instrucción especial (SLEEP en los PIC), que les pasa al estado de reposo o de bajo consumo, en el cual los requerimientos de potencia son mínimos. En dicho estado se detiene el reloj principal y se “congelan” sus circuitos asociados, quedando sumido en un profundo “sueño” el microcontrolador. Al activarse una interrupción ocasionada por el acontecimiento esperado, el microcontrolador se despierta y reanuda su trabajo.[54]

- **Conversor A/D (CAD)**

Los microcontroladores que incorporan un Conversor A/D (Analógico/Digital) pueden procesar señales analógicas, tan abundantes en las aplicaciones. Suelen disponer de un multiplexor que permite aplicar a la entrada del CAD diversas señales analógicas desde las patitas del circuito integrado.

- **Conversor D/A (CDA)**

Transforma los datos digitales obtenidos del procesamiento del computador en su correspondiente señal analógica que saca al exterior por una de las patitas de la cápsula. Existen muchos efectores que trabajan con señales analógicas.

- **Comparador analógico**

Algunos modelos de microcontroladores disponen internamente de un Amplificador Operacional que actúa como comparador entre una señal fija de referencia y otra variable que se aplica por una de las patitas de la cápsula. La salida del comparador proporciona un nivel lógico 1 ó 0 según una señal sea mayor o menor que la otra.

También hay modelos de microcontroladores con un módulo de tensión de referencia que proporciona diversas tensiones de referencia que se pueden aplicar en los comparadores.

- **Modulador de anchura de impulsos o PWM**

Son circuitos que proporcionan en su salida impulsos de anchura variable, que se ofrecen al exterior a través de las patitas del encapsulado.

- **Puertas de E/S digitales**

Todos los microcontroladores destinan algunas de sus patitas a soportar líneas de E/S digitales. Por lo general, estas líneas se agrupan de ocho en ocho formando Puertas.

Las líneas digitales de las Puertas pueden configurarse como Entrada o como Salida cargando un 1 ó un 0 en el bit correspondiente de un registro destinado a su configuración.

- **Puertas de comunicación**

Con objeto de dotar al microcontrolador de la posibilidad de comunicarse con otros dispositivos externos, otros buses de microprocesadores, buses de sistemas, buses de redes y poder adaptarlos con otros elementos bajo otras normas y protocolos. Algunos modelos disponen de recursos que permiten directamente esta tarea, entre los que destacan:

- UART, adaptador de comunicación serie asíncrona.

- USART, adaptador de comunicación serie síncrona y asíncrona
- Puerta paralela esclava para poder conectarse con los buses de otros microprocesadores.
- USB (Universal Serial Bus), que es un moderno bus serie para los PC.
- Bus I2C, que es un interfaz serie de dos hilos desarrollado por Philips.
- CAN (Controller Area Network), para permitir la adaptación con redes de conexión multiplexado desarrollado conjuntamente por Bosch e Intel para el cableado de dispositivos en automóviles. En EE.UU. se usa el J1850.[54]

2.4.7. HERRAMIENTAS PARA EL DESARROLLO DE APLICACIONES

Uno de los factores que más importancia tiene a la hora de seleccionar un microcontrolador entre todos los demás es el soporte tanto software como hardware de que dispone. Un buen conjunto de herramientas de desarrollo puede ser decisivo en la elección, ya que pueden suponer una ayuda inestimable en el desarrollo del proyecto.

Las principales herramientas de ayuda al desarrollo de sistemas basados en microcontroladores son:

- **Desarrollo del software:**
 - **Ensamblador.** La programación en lenguaje ensamblador puede resultar un tanto ardua para el principiante, pero permite desarrollar programas muy eficientes, ya que otorga al programador el dominio absoluto del sistema. Los fabricantes suelen proporcionar el programa ensamblador de forma gratuita y en cualquier caso siempre se puede encontrar una versión gratuita para los microcontroladores más populares.
 - **Compilador.** La programación en un lenguaje de alto nivel (como el C o BASIC) permite disminuir el tiempo de desarrollo de un producto. No obstante, si no se programa con cuidado, el código resultante puede ser mucho más ineficiente que el programado en ensamblador. Las versiones más potentes suelen ser muy caras, aunque para los microcontroladores más populares pueden encontrarse versiones demo limitadas e incluso compiladores gratuitos.
- **Depuración:** debido a que los microcontroladores van a controlar dispositivos físicos, los desarrolladores necesitan herramientas que les permitan comprobar el buen funcionamiento del microcontrolador cuando es conectado al resto de circuitos.
- **Simulador.** Son capaces de ejecutar en un PC programas realizados para el microcontrolador. Los simuladores permiten tener un control absoluto sobre la

ejecución de un programa, siendo ideales para la depuración de los mismos. Su gran inconveniente es que es difícil simular la entrada y salida de datos del microcontrolador. Tampoco cuentan con los posibles ruidos en las entradas, pero, al menos, permiten el paso físico de la implementación de un modo más seguro y menos costoso, puesto que ahorraremos en grabaciones de chips para la prueba in-situ.

- **Placas de evaluación.** Se trata de pequeños sistemas con un microcontrolador ya montado y que suelen conectarse a un PC desde el que se cargan los programas que se ejecutan en el microcontrolador. Las placas suelen incluir visualizadores LCD, teclados, LEDs, fácil acceso a los pines de E/S, etc. El sistema operativo de la placa recibe el nombre de programa monitor. El programa monitor de algunas placas de evaluación, aparte de permitir cargar programas y datos en la memoria del microcontrolador, puede permitir en cualquier momento realizar ejecución paso a paso, monitorizar el estado del microcontrolador o modificar los valores almacenados los registros o en la memoria.

- **Emuladores en circuito.** Se trata de un instrumento que se coloca entre el PC anfitrión y el zócalo de la tarjeta de circuito impreso donde se alojará el microcontrolador definitivo. El programa es ejecutado desde el PC, pero para la tarjeta de aplicación es como si lo hiciese el mismo microcontrolador que luego irá en el zócalo. Presenta en pantalla toda la información tal y como luego sucederá cuando se coloque la cápsula.

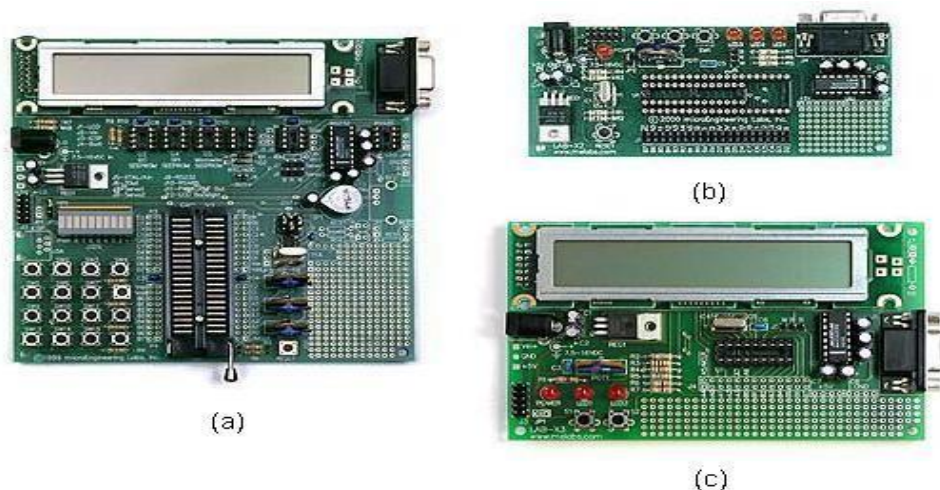


Figura 28: Placas de evaluación: (a) LAB-X1, (b) LAB-X2, (c) LAB-X3, todas las placas son ofrecidas por microEngineering Labs, Inc. (<http://www.melabs.com>)

2.5. BLUETOOTH

El bluetooth, propiamente dicho, es un sistema de transmisión de datos entre dispositivos a corta distancia que no necesita de cables, ya que la transferencia se hace de manera inalámbrica. La banda de transmisión de datos en la que se mueve es la que va entre los 2,4 a 2,48 GHz de amplio espectro, con posibilidad de transmitir hasta 1600 saltos/s. con un total de **79** frecuencias con intervalos de 1Mhz. [62]



Figura 29: SIMBOLO DE BLUETOOTH. [62]

Esto es lo que viene a ser la definición de bluetooth de una manera técnica, podríamos decir que:

“Bluetooth es un sistema de enviar y recibir archivos, música, datos, vídeo, imágenes o incluso voz sin necesidad de cables y a una corta distancia”. [62]

Además a través de la conexión bluetooth, podremos conectarnos de manera inalámbrica a otro dispositivo para compartir la conexión de Internet. [62]

2.5.1. HISTORIA DE BLUETOOTH

El nombre de bluetooth tiene un origen de procedencia nórdica, concretamente del **rey** de Noruega y Dinamarca llamado Harald Blåtand, cuya traducción literal al inglés sería la de *Harold Bluetooth*. Este noble personaje fue conocido por ser un buenísimo comunicador, el cual hizo la gran gesta de unir las diferentes tribus noruegas, suecas y danesas. [62]

2.5.2. PRINCIPIOS DEL BLUETOOTH

La primera empresa que creó un equipo de trabajo para investigar sobre sistemas de comunicación entre dispositivos, fue Ericsson, que allá por el año 1994 comenzó la investigación de una nueva interfaz de bajo consumo y coste, destinada al envío y recepción de datos entre teléfonos móviles y otros dispositivos. [62]

No fue hasta el año 1999, cuando se creó el SIG de Bluetooth (*Special Interest Group*), el cual consistía en la unión de diferentes empresas, entre las cuales se encontraban en un primer momento Ericsson, Intel, Nokia, Toshiba e IBM. A estas empresas fundadoras del grupo, tan solo unos meses después se les unieron otras empresas de la tecnología tan importantes como Microsoft, 3COM, Motorola y Lucent. [62]

2.5.3. VERSIONES DE BLUETOOTH

BLUETOOTH V1.0 Y V1.0B

Los primeros emisores receptores de bluetooth, fueron el v1.0 y v1.0B, los cuales ya están prácticamente obsoletos, y dieron muchísimos problemas a los fabricantes de teléfonos para la interacción entre dispositivos de diferentes compañías, asimismo, tenían el gran defecto que en cada transmisión de datos se enviaba nuestra dirección privada de dispositivo bluetooth, perdiendo así el anonimato que nos pudiese brindar este tipo de conexión inalámbrica. [62]

BLUETOOTH V1.1

- Usa el estándar IEEE 802.15.1-2002
- Corregidos errores de las versiones anteriores.
- Canales no encriptados añadidos y soportados.
- Añadido el indicador de la señal o también denominado (*RSSI*)

BLUETOOTH V1.2

- Compatibilidad con usb 1.1.
- Mejora la velocidad de conexión y transferencia de datos.
- Añadida la función de detección de otros dispositivos bluetooth en el radio de actuación.
- Notables mejoras en la calidad del audio.
- Host Controller Interface (*HCI*)
- Nuevo protocolo estándar IEEE 802.15.1-2005.
- Añadido control de flujo y modos de retransmisión L2CAP.

BLUETOOTH V2.0 + EDR

La mejora implementada en esta nueva versión, hace referencia a la opción del propio fabricante del dispositivo de incorporar la EDR (*Enhanced Data Rate*), esto no viene a significar que todos los dispositivos 2.0 vengan con este sistema de transmisión de datos a mayor velocidad, ya que como he comentado es de carácter opcional. Su transferencia máxima de datos es de 3Mb/s. aunque su tasa real máxima sea la de 2.1Mb/s. esta versión mantiene la compatibilidad con la versión anterior de la interfaz bluetooth. [62]

BLUETOOTH V2.1 + EDR

Las mejoras de esta nueva versión, son mejoras sustanciales siempre mirando hacia la seguridad de nuestros datos, así de esta manera se ha añadido Secure Simple Pairing (SSP), lo que permite un mejor filtrado de nuestros datos y una seguridad superior a la de la versión anterior. A su vez, se ha mejorado notablemente el consumo de energía, gracias a la nueva tecnología *low power*. [62]

BLUETOOTH V3.0 + HS

Este nuevo modelo de la interfaz, fue lanzado en abril de 2009, y su mayor logro es el aumento de la velocidad de transmisión de datos hasta los 24Mb/s., además de incluir una nueva característica la cual hace uso del wifi para el envío y recepción de grandes paquetes de datos, usando el estándar 802.11 de alta velocidad, esta nueva característica es denominada Alternativa MAC / PHY [62]

BLUETOOTH V4.0

Esta versión es la más reciente de todas, y fue lanzada en el año 2010, combina la tecnología bluetooth clásica con la conexión inalámbrica vía wifi, para dotar a los dispositivos en los que vienen instalados de una velocidad de emisión y transferencia de datos de nada más y nada menos que de 32Mb/s.. Esta nueva interfaz de bluetooth viene incluida en los más avanzados Smartphones y dispositivos tecnológicos de última generación. [62]

BLUETOOTH V4.1

Los responsables del desarrollo de la especificación Bluetooth han presentado una nueva versión. La actualización 4.1 que incorpora novedades importantes de cara al usuario al facilitar la reconexión entre sus dispositivos una vez que estos salen y vuelven a entrar en el radio de acción, y que claramente orienta su uso al internet de las cosas.

Estas novedades facilitan la conexión y reconexión de dispositivos, como hemos dicho, sin la necesidad de que el usuario deba hacer nada, al menos una menor frecuencia de interacción. Además permitirá que un dispositivo funcione tanto como periférico y hub de datos a la par, se mejora el soporte para convivir con otros protocolos como LTE, intercambio de datos más eficientes, etc. [62]

BLUETOOTH V4.2

De acuerdo con un comunicado de prensa oficial, la versión 4.2 de la especificación básica Bluetooth salió cerca del final de 2014. Director Ejecutivo del Bluetooth Special Interest Group (SIG) Mark Powell dice que la actualización 4.2 espera continuar haciendo Bluetooth Smart "de la mejor solución para conectar toda la tecnología en su vida. "Bluetooth Core Especificación 4.2 esperanzas para dar a los desarrolladores y fabricantes más oportunidades de usar Bluetooth y construir una mejor experiencia de usuario para sus consumidores. [62]

2.5.4. MÓDULO BLUETOOTH SLAVE (HC-06)

2.5.4.1. DESCRIPCIÓN GENERAL:

El módulo Bluetooth HC-06 es un dispositivo económico y sencillo de utilizar. Se puede utilizar para dar conectividad inalámbrica a través de una interfaz serial TTL entre Microcontroladores (PIC, Arduino) y otros dispositivos como PC, laptops o celulares Smartphone. [63]

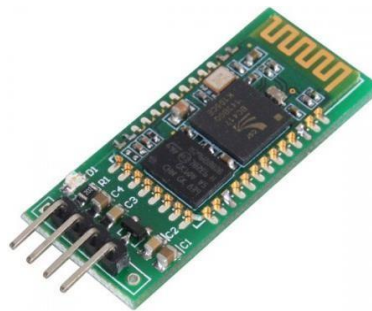


Figura 30: Modulo bluetooth [63]

2.5.4.2. ESPECIFICACIONES:

- Protocolo: Bluetooth v2.0 + EDR
- Frecuencia: Banda ISM de 2,4 GHz
- Modulación: GFSK (Gaussian Frequency Shift Keying)
- Potencia de emisión: 4 dBm, clase 2
- Sensibilidad: -84dBm a 0.1% VER
- Velocidad asíncrona: 2.1Mbps (máx.) / 160 kbps.
- Velocidad síncronos: 1Mbps/1Mbps
- Alcance 10 mts
- Seguridad: Autenticación y encriptación
- Interfaz: Bluetooth - Puerto serie UART TTL
- Suministro de energía: 5VDC 50 mA
- Temperatura de trabajo: $-20 \sim +75^{\circ}\text{C}$
- Dimensión: 26.9mm x 13mm x 2,2 mm [63]

2.6. ANTECEDENTES DEL PROBLEMA

Según **JULIO ALFREDO CORTES RODRIGUEZ**, en su Tesis titulada **“DISEÑO Y CONSTRUCCIÓN DE UN SISTEMA DE ALIMENTACIÓN PARA UN ARREGLO RGB DE TRES LED DE POTENCIA”**, indica lo siguiente:

La iluminación en estado sólido utilizando LED de potencia es una de las nuevas tecnologías de iluminación para fines comerciales y domésticos; sus principales ventajas, respecto a las lámparas incandescentes y fluorescentes, son larga vida útil, eficiencia y tamaño reducido de sus dispositivos. El propósito de este trabajo de tesis es diseñar y construir un sistema de alimentación para un arreglo de tres LED de potencia. Los LED de potencia requieren de una fuente de corriente constante para su correcto funcionamiento, para ello se diseñó un sistema de alimentación conmutado basado en la topología de convertidor CD/CD reductor (buck) en lazo cerrado; para lograr el lazo cerrado del buck se retroalimenta cada una de las corrientes en los LED hacia un microcontrolador (AT90PWM2B), en el que se implementa el algoritmo de control. Para el arreglo LED se utilizaron los colores rojo, verde y azul, con la finalidad de lograr un sistema de iluminación con mezcla de colores utilizando el modelo RGB. La función de atenuación de intensidad luminosa (dimmer) para lograr la mezcla de colores se implementa en el microcontrolador. Durante el diseño y desarrollo del sistema de alimentación se sigue una metodología de desarrollo de sistemas empujados con la

finalidad de establecer y seguir fases bien establecidas que conducen a la obtención de un sistema robusto. Así mismo, se describe el funcionamiento y la configuración del MCU AT90PWM2B, poniendo énfasis en los periféricos utilizados. Finalmente se obtuvo un sistema que puede generar luz blanca y de color, manteniendo una corriente promedio de 700 mA en cada LED

Según **FRANCISCO JAVIER PREYRA PITTA**, en su Tesis titulada “**Generación de luz blanca con LEDs de potencia mediante la técnica RGB**” indica lo siguiente:

Los diodos emisores de luz (LED) tienen una gran popularidad como elementos señalizadores los indicando que el equipo se encuentra encendido. Sin embargo, recientes mejoras en la tecnología de semiconductores ha permitido incrementar la capacidad de manejo de potencia del LED, dando lugar a la creación del LED de potencia. La potencia que puede manejar este tipo de LED puede incrementarse a valores tales que pueden utilizarse en aplicaciones de iluminación. Este hecho puede considerarse como la revolución más grande en iluminación en los últimos años. Los LEDs de potencia generan luz, dependiendo del material con que se fabrican, con diferentes longitudes de onda, es decir, pueden generar distintos colores. Sin embargo, hasta el momento para generar luz blanca se utilizan LEDs de color azul o ultravioleta pasando dicha luz por un filtro a base de fósforos. No obstante su sencillez, el proceso de filtrado reduce la eficacia luminosa. Otro método para generar luz blanca es la combinación de colores Rojo, Verde y Azul, la cual se le denomina método RGB. Esta técnica cuenta con la eficacia luminosa más alta en la producción de luz blanca mediante LEDs de potencia. Sin embargo, es necesario tener en cuenta ciertas características de cada uno de los tres colores. Tales requerimientos hacen necesario incluir un control en lazo cerrado.

El presente trabajo de investigación aborda el diseño de una lámpara de luz blanca basada en LEDs de potencia. La técnica utilizada para la generación de luz blanca es el método RGB. Para la implantación del control se utiliza un controlador Proporcional Integral. Los resultados experimentales muestran buena estabilidad y respuesta tanto en estado estable como en transitorio. Las características cromáticas de la luz generada son: Índice de Rendimiento de Color de 95 y Temperatura Correlacionada de Color de 6000°K. Esto representa una distancia de 0.0083 ΔE_{uv} con respecto a la referencia D65 (referencia de luz blanca estándar); esta variación en la posición del diagrama de cromaticidad no es perceptible por el ojo humano, es decir, se percibe como luz blanca

de excelente calidad. Por otro lado, la tesis también presenta un medidor de calidad de la luz producida por los LEDs: el colorímetro. Cabe señalar que la implementación del colorímetro se llevó a cabo de manera muy simple y de bajo costo.

Según **MARCELO FRANKLIN, GARCIA ALVAREZ**, en su Tesis titulada **“DISEÑO Y CONSTRUCCION DE UN LETRERO DINAMICO BASADO EN LA UTILIZACION DE LEDS RGB”** indica lo siguiente:

La realización del presente proyecto tiene como objetivo primordial la incorporación a nuestro mercado, una nueva tecnología para la elaboración de letreros luminosos. Este avance tecnológico consiste en la utilización de diodos emisor de luz conocidos como “leds”; los mismos que nos permitirán obtener resultados muy exitosos como son contar con un letrero de bajo consumo de energía, amigo del medio ambiente, novedoso, de larga vida útil y con colores extraordinarios. De acuerdo a lo mencionado y con el afán de hacer partícipes de la mejor manera de la introducción de innovadora tecnología en nuestra ciudad el desarrollo del presente proyecto de tesis, ha sido enfocado hacia 5 capítulos, siendo los siguientes: Capítulo I: Diodos emisores de luz; en este capítulo se da a conocer su historia, sus primeros usos, características, beneficios, hasta el moderno uso que se les puede dar en la actualidad. Capítulo II: Controlador especializado para el led; este capítulo versa sobre el chip que se encarga de manejar y emitir las señales de control para los leds, se realiza un profundo análisis de su arquitectura, cualidades, características, funcionamiento, e interfaz. Capítulo III: Microcontrolador PIC18f452; en el desarrollo de este capítulo se da a conocer las principales funciones y características de este chip, sin entrar en un profundo análisis debido que este chip es ya muy conocido, utilizado y existe mucha información sobre el mismo. Capítulo IV: Diseño y elaboración de los circuitos electrónicos; este capítulo es uno de los más importantes para el lector, debido a que es en el que consta, el diseño electrónico de cada una de las partes, todos los esquemas y explicaciones del funcionamiento, su ensamblaje, programación de los chips, la construcción física de letrero, montaje de los leds, resultado de las pruebas de funcionamiento, y las fotografías de proyecto tanto de las partes más importantes como del proyecto final en funcionamiento. Capítulo V: Interfaz con el usuario; este capítulo versa sobre el manejo del letrero, como realizar la ejecución de un programa u otro, también se detalla la descripción de cada uno de los programas que se pueden ejecutar.

2.7. ELABORACIÓN DE LA HIPÓTESIS

Las hipótesis indican lo que estamos buscando o tratando de probar y se definen como explicaciones tentativas del fenómeno investigado, formulado manera de proposiciones (Hernández y otros 2003). En consecuencia, constituye el eje principal de orientación que debe contrastarse con la realidad estudiada.

En concordancia con la formulación del problema y el objetivo planteado, la hipótesis de investigación concreta a esta situación problemática, queda enunciada bajo la siguiente proposición:

Si es importante y posible realizar diseño e implementación de un prototipo de lámpara de fototerapia led con control de luminosidad para el tratamiento de ictericia en neonatos en nuestro medio.

CAPÍTULO 3: METODOLOGÍA

3.1. TIPO DE ESTUDIO

De acuerdo con el propósito de la investigación, la naturaleza de la problemática y el objetivo formulado, la investigación será técnica. Según Velásquez y Rey este tipo de investigación tiene como finalidad la solución de problemas prácticos, lo cual implica intervención o modificación de la propia realidad; se pretende demostrar la utilidad de la intervención que se realizará.

El desarrollo de la presente investigación se basará en el método de la ciencia o, simplemente, método científico; el mismo que está conformado por una serie de actividades correctamente articuladas que conducen a un resultado coherente con la propuesta de solución. Tales actividades son: problema nuevo para la ciencia, objetivos de la investigación, hipótesis de la investigación, variables, universo y muestra, técnicas e instrumentos, aplicación de los instrumentos, análisis de la información, contrastación de la hipótesis, formulación de conclusiones y recomendaciones,[56].

Por responder más apropiadamente a la naturaleza y tipo de trabajo realizado, se ha seleccionado un diseño de *Investigación preexperimental con preprueba*. Este diseño podría diagramarse de la siguiente manera:

G O1 X O2

Consiste en realizar una medición previa O1 en la unidades de análisis; administrar un estímulo o tratamiento a un grupo y después realizar una segunda medición en una o más variables para observar cuál es el nivel del grupo en estas variables, [57].

G será “Lámpara de Fototerapia Led” X (Tarjeta Electrónica con control de potencia en Panel de leds formada por una tira leds RGB de 5metros) luego se describirá el nuevo sistema.

3.2. COBERTURA DEL ESTUDIO

3.2.1. POBLACIÓN

Para el desarrollo del presente estudio se toma en cuenta los clientes o padres de neonatos con Ictericia a los que se le aplicara el nuevo sistema de fototerapia en la ciudad de Piura.

3.2.2. MUESTRA

La muestra básicamente será un determinado número de neonatos de la ciudad de Piura.

3.3. TÉCNICAS E INSTRUMENTOS

3.3.1. TÉCNICAS

- Observación y análisis del proceso de control de potencia en el panel de leds conformada por una tira leds RGB de 5 metros.
- Técnicas de rediseños de procesos (rediseño del proceso de control de potencia en el panel de leds).

3.3.2. INSTRUMENTOS

- Guía de observación.
- Prototipo del sistema.

CAPÍTULO 4:

DISEÑO DE TARJETA ELECTRONICA PARA LAMPARA DE FOTOTERAPIA LED CON CONTROL DE LUMINOSIDAD

4.1. DIAGRAMA DE BLOQUES DE LAMPARA LED Y TARJETA ELECTRONICA

El problema que se plantea en la presente Tesis es Diseñar una lámpara led y tarjeta electrónica que permita controlar la potencia de la luminosidad de los leds para lo cual se usara Hardware y Software Libre, la tarjeta o hardware principal será ARDUINO UNO, se seleccionara los leds cantidad y sus características, también se diseñara el driver de potencia y panel de control de luminosidad, se usara sensores de distancia para determinar distancia horizontal y vertical con la Incubadora y paciente

En la Figura 31 se muestra el diagrama de bloques general de la lámpara led donde se tiene que controlar su posición mediante un Servomotor, la potencia de la luminosidad y tener en cuenta los datos de las distancias de los tres sensores de ultrasonido.

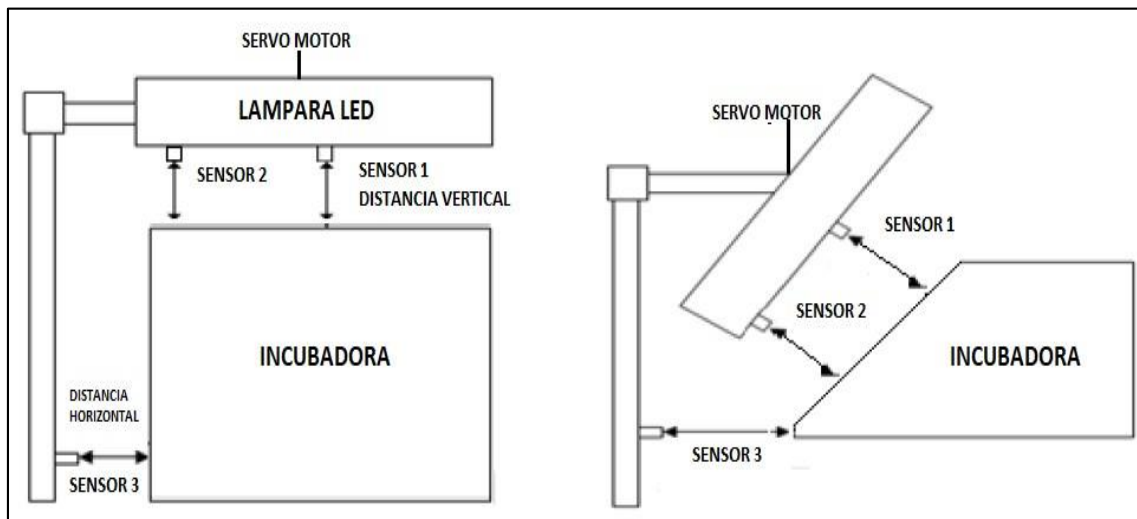


Figura 31: Diagrama de bloques de lámpara para fototerapia led.

En la Figura 32 se muestra el diagrama de bloques de la tarjeta electrónica que controlara la Lámpara led de Fototerapia. Esta tarjeta principalmente se encargara controlar la luminosidad de los leds, la posición de la lámpara con servomotor y medición de distancia de tres sensores de ultrasonido.

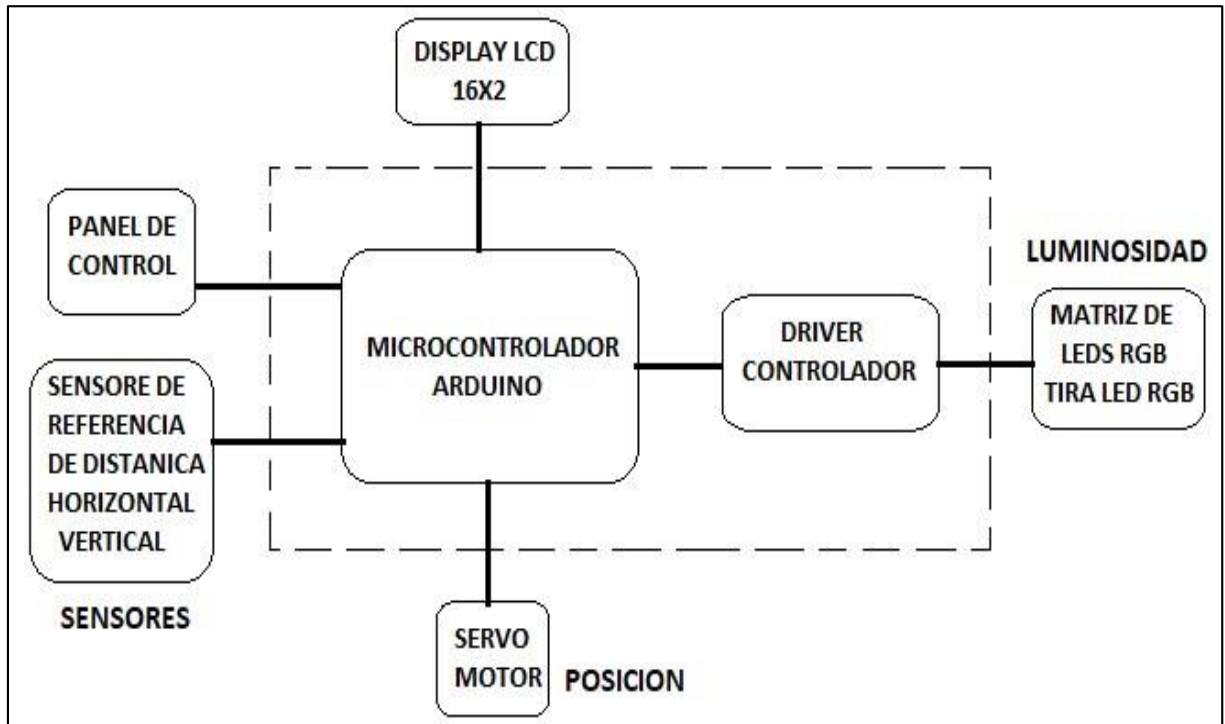


Figura 32: Diagrama de bloques de tarjeta electrónica para control de lámpara de fototerapia led.

El hardware de la Figura 31 y Figura 32. Tiene 7 etapas bien definidas, que se definen a continuación:

- Luminosidad
- Driver controlador de Potencia de leds
- Posición de lámpara por Servomotor
- Sensores de distancia
- Panel de control
- Display LCD
- Microcontrolador – ARDUINO UNO.

4.2. DESCRIPCION DE LAS ETAPAS DEL DISEÑO DE LAMPARA LED DE FOTOTERAPIA.

4.2.1. ETAPA DE LUMINOSIDAD

4.2.1.1. ÁREA DE ESTUDIO

Para el diseño de la lámpara de fototerapia se debe conocer el objeto y el área que va a recibir la radiación lumínica. El objeto que va a ser irradiado en este caso es el neonato (recién nacido a término o pre término) y el área de estudio es la incubadora en la que va a reposar el recién nacido durante la fototerapia.

Para determinar el valor del área de estudio se debe conocer las medidas más comunes que poseen incubadoras en el mercado (ver tabla 7).

INCUBADORAS				
Modelo	Largo (cm)	Ancho (cm)	Área (cm ²)	Altura (cm)
DragerCaleoIncubator	116.7	68.7	8017.29	122 - 152
AIRSHIELDS® C-300	81.3	55.9	4544.67	128.9
Air-Shields® Transport Incubator Ti100	102	56.5	5763	88.3 - 118.7
Ohmeda® Care-Plus™ Incubator	65	34.8	2262	135.9
Airshields® 500 Transport Incubator	95.9	52.7	5053.93	50.8
GE Care Plus Access Series 3000/4000 Incubators	83.8	71.1	5958.18	135.9

Tabla 7: Características de incubadoras más comerciales [49]

De igual manera se debe conocer el espacio que ocupa el neonato dentro del área de estudio (ver tabla 8).

EDAD	TALLA NIÑOS(cm)	TALLA NIÑAS (cm)	TORSO (cm)
Recién Nacido	50.3	50.3	20
3 meses	60	59	20
6 meses	67	65	21
9 meses	72	70	21
12 meses	76	74	22
15 meses	79	77	22

Tabla 8: Tabla orientativa de las medidas ideales para niños y niñas por edad [50]

Como ya se mencionó en el capítulo 2, los más propensos a desarrollar hiperbilirrubinemia son los recién nacidos (de 2 a 5 días de edad), pero esto no significa que bebés de mayor edad desarrollen esta patología. La fototerapia debe poseer luz

uniformemente distribuida por toda la superficie corporal del paciente, es por esto que para el estudio lumínico se tomará las medidas de un bebé de 15 meses (según la tabla 8) y se las adjudicará al objeto de estudio.

Si se realiza una media de las medidas de largo y ancho de la tabla 7 se obtiene un valor promedio para el área de estudio, el promedio del largo es 85 cm y el promedio para el ancho es 60 cm. Analizando las medidas del bebé de 15 meses de edad con los brazos abiertos (como se muestra en la figura 33) se tiene valores de largo y ancho de 79 y 56 cm respectivamente.

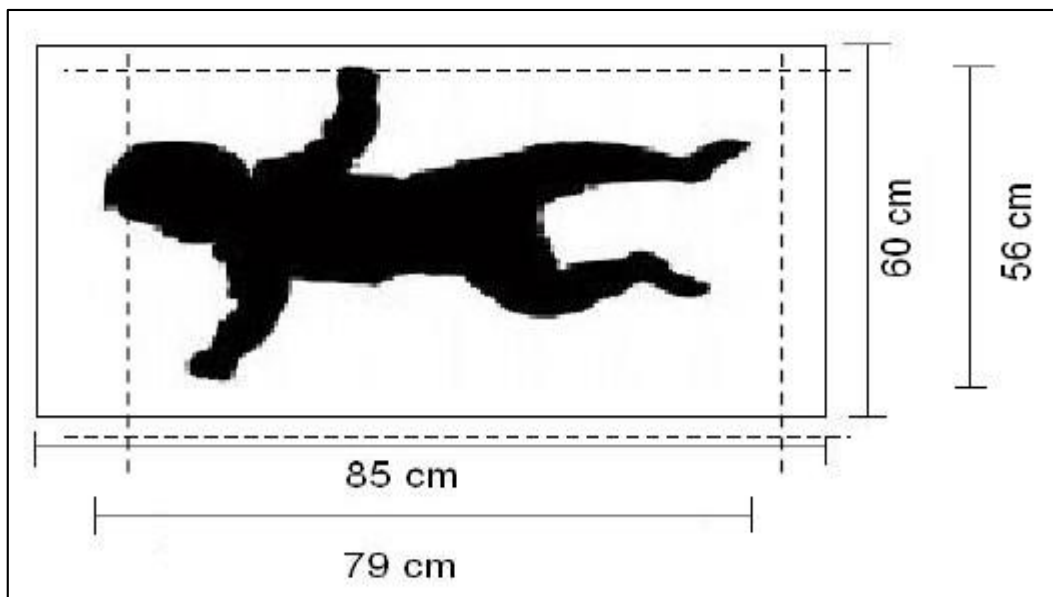


Figura 33: Área de estudio

Como se observa en la figura 33 el área que ocupa el objeto de estudio es casi igual al área de estudio obtenida, en la que se debe centrar toda la irradiancia de la fuente (10 - 30 uW/cm²/nm). Dentro del área de estudio es muy importante que el espectro lumínico sea totalmente uniforme y cubra toda el área para mayor eficiencia del tratamiento de la fototerapia en caso de movilidad del paciente. La lámpara de fototerapia que se va a construir deberá poseer una irradiancia mínima de **10 a 30uW/cm²/nm** con una longitud de onda pico de **458 nm**, que distribuya uniformemente la irradiancia espectral en un área de $85 \times 60 = 5100 \text{ cm}^2$.

4.2.1.2. CÁLCULO LUMÍNICO

Anteriormente se definió que la radiación de energía lumínica para la fototerapia debe ser de mayor o igual **10 uW/cm²/nm** para tratamiento normal y mayor o igual a **30 uW/cm²/nm** para tratamiento intensivo (según la Academia Internacional de Pediatría). Para fines experimentales se utilizará una irradiancia espectral de **22 a 45 uW/cm²/nm** en el prototipo de lámpara para fototerapia; estos valores son propuestos observando la Tabla 9. Tomado como referencia el modelo de equipo **LULLABY LED PT**. Los valores de irradiancia espectral propuestos están dentro de los intervalos probados y aceptados en la medicina.

EQUIPOS DE FOTOTERAPIA		
Modelo	Longitud de onda (nm)	Radiancia Espectral (uW/cm ² /nm)
BabyBlue	450 - 470	10 - 30
NeoBlue	450 - 595	12 - 35
Bilitron	400 - 550	4 - 50
Bililed	453 - 487	10 - 50
Dräger lámpara de fototerapia 4000	460- 550	28.8 - 41.4
Bili-therapy spot type	460 - 490	30 - 40
B-200 InfantPhototherapyUnit	400 - 760	500
BT - 400 Phototherapy	460 - 470	> 30
Lámpara de fototerapia Cobams	420 - 480	> 10
Naos and Naos Plus	420 - 480	< 60
Lullaby Led PT	450 - 465	22 - 45

Tabla 9: Longitud de onda e Irradiación espectral de equipos de fototerapia [51]

Es muy importante conocer conceptos básicos de radiometría y fotometría para el estudio lumínico.

Radiometría: Rama de la física dedicada a la medida de las radiaciones electromagnéticas.

Fotometría: Parte de la radiometría que se ocupa de la parte del espectro electromagnético que el ojo humano percibe como sensación luminosa.

El objetivo de la radiometría y la fotometría es medir la radiación, propagación y recepción de la energía electromagnética. Cuando se conoce una magnitud espectral para cada valor de la longitud de onda (X_λ) y se quiere determinar el valor neto de la magnitud radiométrica o fotométrica (X) correspondiente para un rango determinado de longitudes de onda (λ_1, λ_2) se debe integrar.

$$X = \int_{\lambda_1}^{\lambda_2} X_{\lambda} d\lambda \quad (5)$$

Todas las magnitudes radiométricas y su magnitud fotométricas correspondiente se resumen en la tabla 10.

J	W	sr	V(λ)	lm	lx	nt	cd	A	τ	t	Ω
julio	Vatio (watt)	Estéreo radián	Eficiencia luminosa	lumen	lux	nit	candela	área	volumen	tiempo	Ángulo sólido
Magnitudes Radiométricas (subíndice “e” ⇔ “energética”)						Magnitudes Fotométricas (subíndice “v” ⇔ “visión”)					
Magnitud	Símbolo (unidad)	Definición		Magnitud	Símbolo (unidad)	Definición					
Energía Radiante	Q_e $J=W \cdot s$	—		Energía Luminosa	Q_v $lm \cdot s$	—					
Densidad de energía radiante	W_e J/m^3	$W_e = dQ_e/d\tau$		Densidad de energía luminosa	W_v $(lm \cdot s)/m^3$	$W_v = dQ_v/d\tau$					
Flujo (o potencia) radiante	Φ_e W	$\Phi_e = dQ_e/dt$		Flujo (o potencia) luminosa	Φ_v lm	$\Phi_v = dQ_v/dt$					
Excitancia radiante o emitanciar.	M_e W/m^2	$M_e = d\Phi_e/dA$ (A=emisor)		Excitancia luminosa o emitanciar.	M_v lm/m^2	$M_v = d\Phi_v/dA$					
Incidencia radiante o Irradiancia	E_e W/m^2	$M_e = \Phi_e/dA$ (A=receptor)		Iluminancia o iluminación	E_v $lm/m^2 = lx$	$M_v = d\Phi_v/dA$					
Intensidad radiante	I_e W/sr	$I_e = d\Phi_e/d\Omega$		Intensidad luminosa	I_v $lm/sr = cd$	$I_v = d\Phi_v/d\Omega$					
Radiancia	L_e $W/(sr \cdot m^2)$	$L_e = dI_e/(dA \cdot \cos\theta)$ (A=emisor)		Luminancia (L) o brillo (B)	L_v $cd/m^2 = nt$	$L_v = dI_v/(dA \cdot \cos\theta)$					
Exposición	H J/m^2	$H = E_e \cdot t$		Exposición	H $lx \cdot s$	$H = E_v \cdot t$					

Tabla 10: Magnitudes radiométricas y fotométricas y su nomenclatura [52]

Las magnitudes radiométricas y fotométricas poseen una relación para pasar de una magnitud a la otra basada en la sensibilidad del ojo humano y la cantidad de luz incidente en él, es decir, la eficiencia luminosa (ver figura 34).

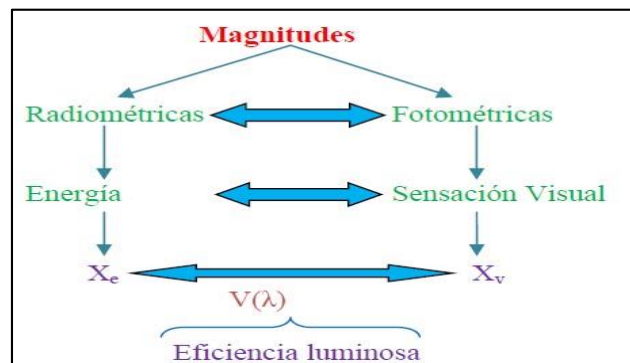


Figura 34: Relación entre magnitudes radiométricas y fotométricas [53]

El valor de una magnitud fotométrica (X_v) es proporcional al de la magnitud radiométrica (X_e) asociada para una longitud de onda dada es:

$$X_v = K(\lambda)X_e \quad (6)$$

La eficacia luminosa espectral $K(\lambda)$ mide la sensibilidad relativa del ojo del observador humano medio para cada longitud de onda. La dependencia de la eficacia luminosa con la longitud de onda se expresa de la siguiente manera:

$$K(\lambda) = K_m V(\lambda) \quad (7)$$

La eficiencia luminosa espectral depende de la cantidad de iluminación que exista en un ambiente para que aumente o disminuya la sensibilidad relativa del ojo humano. Por esta razón la sensibilidad del ojo se analiza como: visión diurna o fotópica y visión nocturna o escotópica.

Visión fotópica $V(\lambda)$: Es la forma normal de visión del ojo humano cuando la iluminación es suficientemente alta ($L_v \geq 3 \text{ cd.m}^{-2}$ en el campo de visión)

$k_m = 683 \frac{\text{lm}}{\text{w}}$ Eficacia espectral luminosa máxima $\lambda_m = 555 \text{ nm}$ Valor máximo que alcanza la eficiencia luminosa

Visión Escotópica $V'(\lambda)$: Es la característica de visión del ojo humano con niveles muy bajos de iluminación ($L_v \leq 3 \times 10^{-5} \text{ cd.m}^{-2}$)

$k'_m = 1700 \frac{\text{lm}}{\text{w}}$ Eficacia espectral luminosa máxima $\lambda'_m = 507 \text{ nm}$ Valor máximo que alcanza la eficiencia luminosa
--

La visión fotópica y escotópica son magnitudes adimensionales normalizadas, es decir, toman valores comprendidos entre 0 y 1 (ver figura 35).

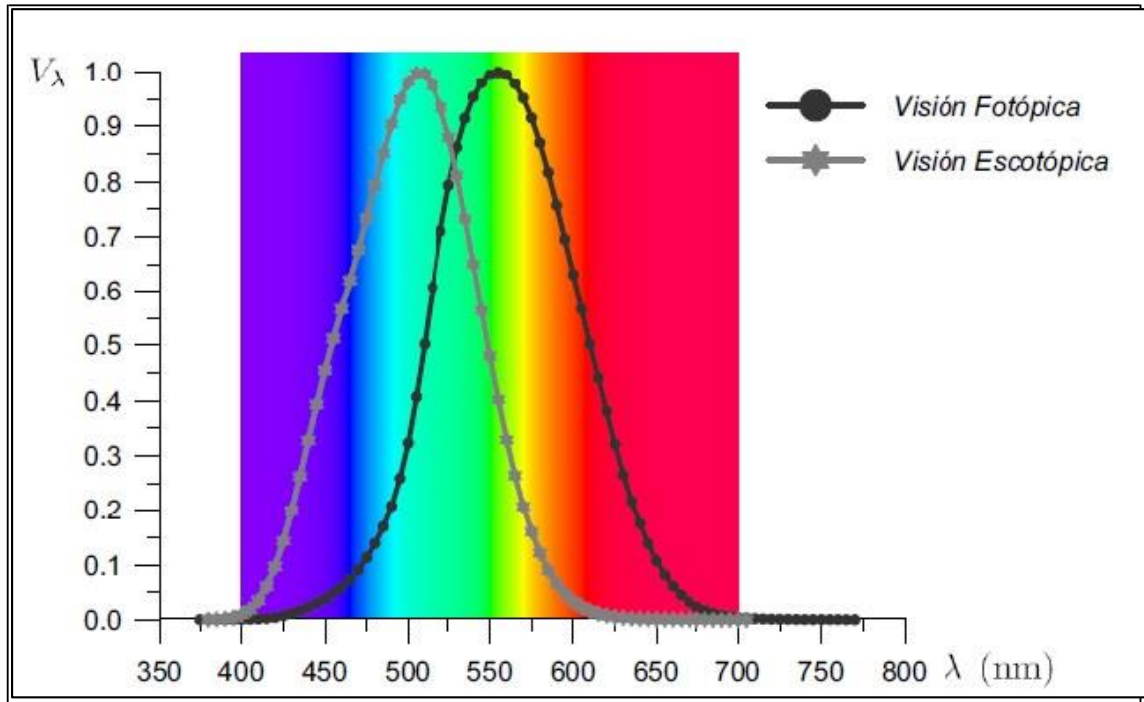


Figura 35: Curvas C.I.E. de la eficiencia luminosa espectral [53]

En el proyecto el objetivo no es la iluminación de un ambiente, es la radiación espectral que pueda irradiar la fuente para la máxima penetración de la luz en la piel. Por lo antes mencionado se utilizará la curva de la visión fotópica para llegar al valor fotométrico correspondiente para el proyecto.

La curva de la visión fotópica viene dada por:

$$V(\lambda) = e^{\left[\frac{-0.5(\lambda-555)^2}{1755.61}\right]} \quad (8)$$

Cuando se tiene una radiación policromática caracterizada por una cierta magnitud radiométrica espectral ($X_e\lambda$) el valor de la magnitud fotométrica espectral correspondiente es:

$$X_{v\lambda} = K(\lambda)X_{e\lambda} \quad (9)$$

El valor neto de la magnitud fotométrica se obtiene con la ecuación 5, si $X=X_v$ se tiene:

$$X_v = \int_{\lambda_1}^{\lambda_2} X_{v\lambda} d\lambda \quad (10)$$

Reemplazando la ecuación 9 en 10 se obtiene:

$$X_v = \int_{\lambda_1}^{\lambda_2} K(\lambda) X_{e\lambda} d\lambda \quad (11)$$

A saber que la eficacia luminosa espectral depende de la curva de la visión fotópica se reemplaza los valores de la eficacia luminosa espectral máxima, la ecuación 7 y 8 en la ecuación 11.

$$X_v = \int_{\lambda_1}^{\lambda_2} \left(683 \frac{lm}{W} \right) \left(e^{\left[\frac{-0.5(\lambda-555)^2}{1755.61} \right]} \right) X_{e\lambda} d\lambda \quad (12)$$

Como ya se tiene definido el valor de la irradiancia espectral ($E_{e\lambda}$), ahora se debe encontrar su magnitud fotométrica correspondiente que es la iluminancia (E_v). En la ecuación 5 se definió que X representa a las magnitudes fotométricas y X_λ representa a las magnitudes radiométricas espectrales, por lo que se procederá definir la magnitud con la que se trabajará en la ecuación 12, así:

$$E_v = \int_{\lambda_1}^{\lambda_2} \left(683 \frac{lm}{W} \right) \left(e^{\left[\frac{-0.5(\lambda-555)^2}{1755.61} \right]} \right) E_{e\lambda} d\lambda \quad (13)$$

El Led es una fuente lumínica muy especial, se la puede considerar como una fuente emisora de luz monocromática (para los colores RGB) y también como una fuente emisora de luz policromática. Para éste caso se lo considerará como una fuente de luz monocromática y policromática a la vez, entonces la ecuación 13 quedará de la siguiente forma:

$$E_v = 683 \frac{lm}{W} \cdot E_{e\lambda} \cdot (\lambda_2 - \lambda_1) \int_{\lambda_1}^{\lambda_2} e^{\left[\frac{-0.5(\lambda-555)^2}{1755.61} \right]} d\lambda \quad (14)$$

La irradiancia espectral debe ser constante para todas las longitudes de onda del espectro de la fuente, y es por esto que no se la integrará conjunto la curva de la visión fotópica.

El espectro es una característica implícita en la fuente que no se la puede controlar, así que ese dato se debe consultar en el catálogo de la fuente que se va a utilizar. El único requisito que se debe establecer para la búsqueda de la fuente es que posea una longitud de onda máxima o dominante que se encuentre cerca de los 460 nm [15].

Después de una larga búsqueda en proveedores en todo el mundo se encontró el Led con las siguientes características (ver tabla 11):

Color de Emisión	Corriente inversa (mA)	Consumo de energía (watt)	Longitud de onda (nm)	Controlador de Voltaje (V)	Flujo Luminoso (lm)	Ángulo de emisión (°)
Azul	700	2.38	460 - 465	3.0 - 3.4	150 - 220	120

Tabla 11: Características del led.

Conociendo λ_1 y λ_2 la ecuación 14 queda de la siguiente forma:

$$E_v = 683 \frac{lm}{W} \cdot 0.45 \frac{W}{nm \cdot m^2} \cdot (465 nm - 460 nm) \int_{460}^{465} e^{\left[\frac{-0.5(\lambda-555)^2}{1755.61} \right]} d\lambda$$

Para poder conocer la iluminancia que debe tener la lámpara de fototerapia resolvemos la ecuación 14 con el valor máximo y mínimo de irradiancia espectral (22 - 45uW/cm²/nm); los resultados son:

$$E_{max} = 6744.04 \text{ lx}$$

$$E_{min} = 3292.2 \text{ lx}$$

4.2.1.3. DISTRIBUCIÓN LUMÍNICA

El Led que se va a utilizar para el presente proyecto tiene un flujo luminoso máximo de 220 lm y un ángulo de emisión de 120°, para calcular la intensidad luminosa máxima se debe conocer la unidad de ángulo sólido que tiene el Led.

$$\Omega = 2\pi \left(1 - \cos \frac{\alpha}{2} \right) \quad (15)$$

$$\Omega = 2\pi \left(1 - \cos \frac{120^\circ}{2} \right)$$

$$\Omega = \pi \text{ (sr)}$$

Como ya se indicó en el capítulo 2 la unidad del ángulo sólido es el estereorradián (sr) y el ángulo sólido para nuestro Led es de π sr, el cual nos servirá para calcular la intensidad luminosa máxima.

$$I_v = \frac{\Phi_v}{\Omega} \quad (16)$$

$$I_v = \frac{180 \text{ lm}}{\pi \text{ sr}}$$

$$I_v = 57.29 \text{ (cd)}$$

La iluminancia del Led sobre un área específica (0.51 m²) es inversamente proporcional a la distancia entre la fuente y el área. La altura máxima que poseerá la lámpara de fototerapia con respecto al área de estudio es de 45 cm.

$$E_{vtotal} = \sqrt{E_H^2 + E_V^2} \quad (17)$$

$$E_H = \frac{I_v \cdot (\cos \alpha)^3}{h^2} \quad (18)$$

$$E_V = \frac{I_v \cdot (\cos \alpha)^2 \cdot \sin \alpha}{h^2} \quad (19)$$

La iluminancia máxima de Led será cuando el ángulo de visión sea igual a 0° y la iluminancia mínima será cuando el ángulo de visión sea 60° (ver figura 36).

$$\alpha = 0^\circ$$

$$E_H = \frac{(57.29 \text{ cd}) \cdot (\cos 0)^\alpha}{(0.45 \text{ m})^2} = 282.94 \text{ lx}$$

$$E_V = \frac{(57.29 \text{ cd}) \cdot (\cos 0)^\alpha \cdot \sin 0}{(0.45 \text{ m})^2} = 0 \text{ lx}$$

$$E_{vtotal} = \sqrt{(282.94)^2 + (0)^2} = 282.94 \text{ lx}$$

$$\alpha = 60^\circ$$

$$E_H = \frac{(57.29 \text{ cd}) \cdot (\cos 60)^\alpha}{(0.45 \text{ m})^2} = 35.31 \text{ lx}$$

$$E_V = \frac{(57.29 \text{ cd}) \cdot (\cos 60)^\alpha \cdot \sin 60}{(0.45 \text{ m})^2} = 61.25 \text{ lx}$$

$$E_{vtotal} = \sqrt{(35.31)^2 + (61.26)^2} = 70.69 \text{ lx}$$

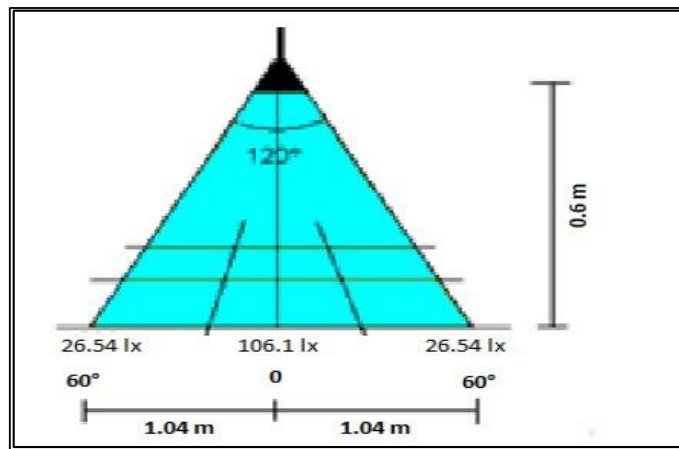


Figura 36: Iluminancia del LED

En este estudio lumínico se llegó a determinar la iluminancia máxima del Led, la cual depende del ángulo del observador y la distancia entre fuente y área. Con este valor se puede calcular el número de Leds que se necesitan para lograr la iluminancia requerida para el sistema.

$$\#_{led} = \frac{E_{max}}{E_{led}} \cdot \frac{A}{1 \text{ m}^2} \quad (20)$$

$$\#_{led} = \frac{(6744.04 \text{ lx})(0.51 \text{ m}^2)}{(282.94 \text{ lx})(1 \text{ m}^2)}$$

$$\#_{led} = 12.15$$

$$\#_{led} = 12 \text{ Led}$$

Con la ecuación 20 se determina el número de fuentes de luz necesarias para ofrecer la iluminancia requerida para irradiar un área específica, tomando en cuenta que:

E_{max} es la iluminancia necesaria para obtener la irradiancia espectral adecuada para la fototerapia.

E_{led} es la iluminancia máxima que puede ofrecer el Led.

A: Es el área específica donde se va a concentrar la energía lumínica de la lámpara.

Se necesita 12 Leds para alcanzar la iluminancia requerida para un área de 0.51 m², ahora el objetivo es plantear una distribución que permita dispersar la iluminancia requerida uniformemente sobre toda el área.

4.2.1.4. DISEÑO DE LA MATRIZ DE LEDS PARA LAMPARA DE FOTOTERAPIA

En el diseño de la matriz de Leds se debe considerar el área que va a ser irradiada lumínicamente y la fuente luminosa que se va a utilizar para construir la matriz. En el punto anterior, las características de la fuente luminosa que se va a utilizar en este proyecto (tabla 4.5) y el área que será irradiada ya fueron definidas. Con estos datos se debe encontrar la forma que debe tener la matriz de Leds y la posición en que debe estar cada Led para obtenerla radiación lumínica deseada.

Hay muchas formas geométricas que se pueden considerar para el diseño de la matriz de Leds. Las formas más comunes que se pueden encontrar en el mercado son: circular (bombillas Led), cuadrada (reflectores Led) y rectangular (lámparas fluorescentes Led). Para encontrar la forma que reproduzca de manera eficiente la radiación de la matriz se debe simular las opciones antes mencionadas y encontrar la distribución idónea.

Realizando simulaciones de varias configuraciones de distribuciones luminosas en el programa Matlab 2010 se concluye que entre mayor distancia exista entre Leds mayor será el área de pico de iluminancia, pero menor será el valor de pico.

Con base a esto se propone un diseño de distribución luminosa rectangular, el cual debe tener mayor distancia entre Leds para aumentar el área de iluminancia y mayor cantidad de Leds para aumentar el pico de iluminancia, ver figura 37.

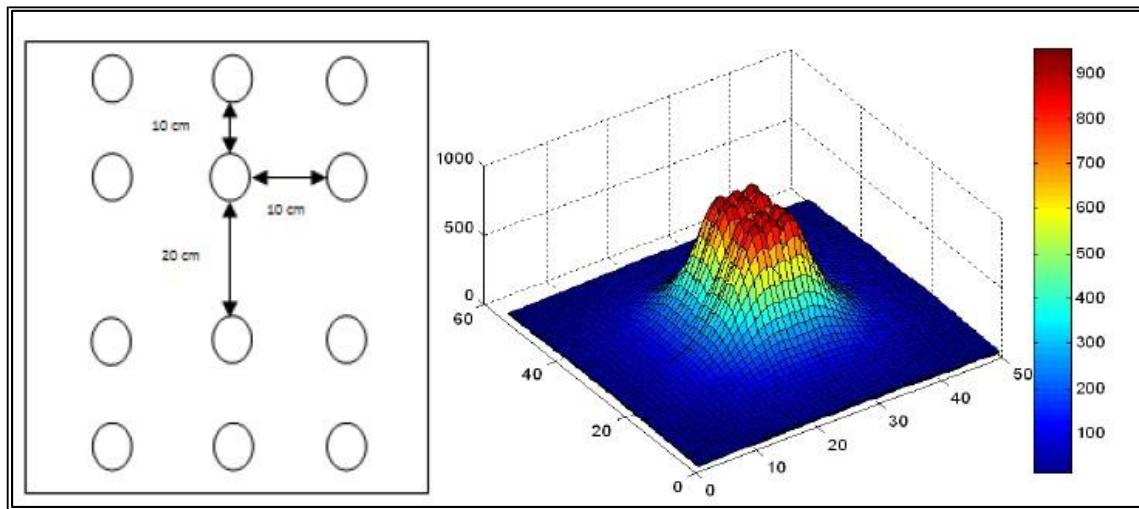


Figura 37: Simulación de la distribución rectangular de Leds en MatLab 2010

En esta distribución se caracteriza por que consta de dos grupos de seis Leds separados por una distancia de 20 cm y una distancia de 10 cm entre Leds. Con la distribución rectangular los requerimientos que necesita el sistema se cumplen en su mayor parte (ver figura 38.).

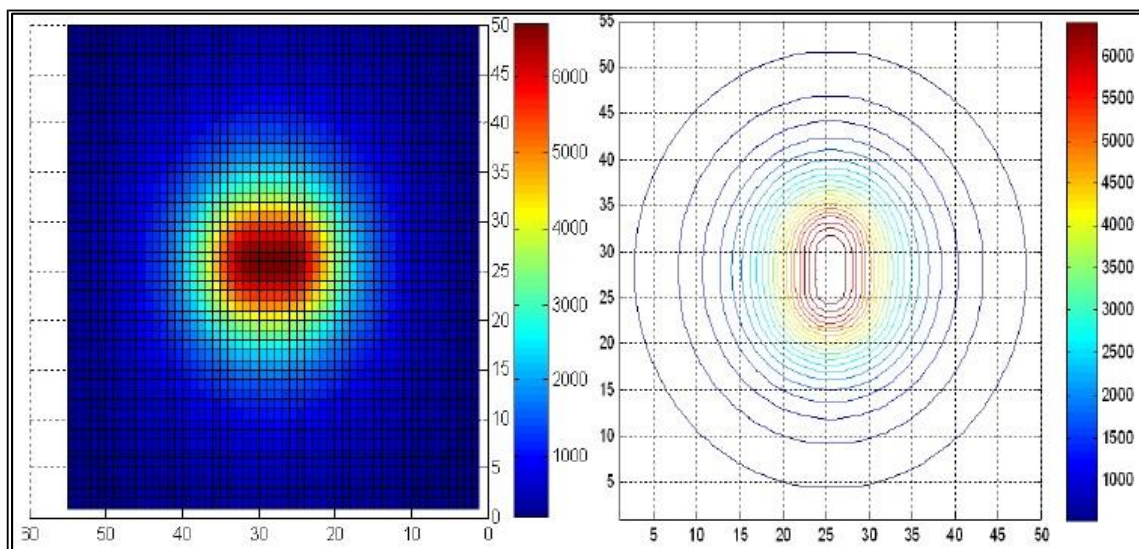


Figura 38: Simulación de niveles de iluminancia (lx) en escala de colores de la distribución

En la figura 38 se observa que el pico máximo que tiene ésta distribución es de 7000 lx cuya área es el anillo más pequeño de la figura y su área es de 24 x 15 cm, el anillo que

le sigue en tamaño tiene una iluminancia de 6500 lx y un área de 30 x 20 cm y el siguiente anillo tiene una iluminancia de 6000 lx y área de 38 x 25 cm.

Con estos primeros datos se comprueba que más del 50% del área de estudio va a tener una iluminancia de 6500 lx (valor necesario para la fototerapia intensiva, 45uW/cm²/nm) y se comprueba también que la distribución rectangular es la adecuada para la lámpara de fototerapia.

Definida la forma física que deberá debe implementar la Matriz de Leds y se procede a diseñar los circuitos electrónicos y la conexión eléctrica que permitan el funcionamiento deseado para el proyecto.

4.2.2. DRIVER CONTROLADOR DE POTENCIA DE LOS LEDS

En el mercado local no se encuentra un led como el especificado en la Tabla 11, por lo tanto utilizaremos Tiras Leds las cuales si hay disponibles a la venta en el mercado local.

Las tiras de LEDs o LED strips son dispositivos de iluminación pre ensamblados. Los LEDs y los componentes auxiliares se encuentran montados sobre un circuito impreso flexible que hace posible su adaptación a casi cualquier superficie. Una de sus caras cuenta con un material autoadhesivo 3M que permite instalar las tiras de forma fácil y rápida. No generan calor ni dañan las superficies donde son aplicadas, Gozan de una larga vida útil por lo que tienen un costo de mantenimiento muy bajo.

Las dimensiones de las tiras más comunes son de 8mm a 10mm de ancho, 3mm de alto y 5 metros de largo. Las tiras pueden ser cortadas cada 3 LEDs, según el modelo el corte será cada 5cm o 10 cm y pueden unirse nuevamente simplemente conectando o soldando sus contactos de cobre ubicados en los extremos de cada segmento.

Las tiras más comunes están preparadas para ser alimentadas con una tensión de 12V CC y en el mercado también hay disponibles tires de LEDs aptas para su conexión a 24V CC y 220 V CA. [59]

4.2.2.1. TIPOS DE TIRAS DE LEDS

Existen en el mercado una gran variedad de cintas de LEDs, cada una satisface una o varias necesidades en el diseño de la instalación. Los principales parámetros que las caracterizan son:

- Tensión de alimentación
- Tipo de LED

- Cantidad de LEDs por metro
- Consumo eléctrico
- Cantidad de luz que emite
- Color
- Grado de protección
- Vida útil

Además de estos parámetros principales, existen varios otros más finos que en primera instancia pueden obviarse pero se hacen necesarios para obtener resultados finales profesionales. Entre ellos:

- Metros de tira por rollo
- Ángulo de iluminación
- Color de la base el circuito impreso o PCB: Generalmente blanco, negro o cobre
- Disposición de los LEDs en una fila o dos filas.
- Certificaciones de calidad: CE / RoHS.
- Peso por metro o por rollo.

Pasamos a describir cada una de las principales características de las tiras de LEDs:

TENSIÓN DE ALIMENTACIÓN

La gran mayoría de las tiras de LEDs funcionan con 12V CC o 24 V CC por lo que para su alimentación desde la red eléctrica será necesario el uso de una fuente de alimentación adecuada.

En el caso de su utilización en vehículos o alejadas de la red, se pueden alimentar directamente con tensión continua desde las baterías. En este caso se verificará que sean compatibles los valores de tensión de las baterías y de las cintas de LEDs. [59]

TIPO DE LED

Los LEDs y los otros componentes usados en estas tiras utilizan tecnología de montaje superficial (SMD o surface mounted device). Se trata de componentes encapsulados en una resina semirígida y que se ensamblan de manera superficial. En el caso de estos LEDs, el encapsulado es del tipo PLCC (Plastic Leadless Carrier Chip). [59]



Figura 39: Tipos de Led 1 [59]

En las tiras más comunes se usan dos tipos de LEDs el PLCC 5050 (5,0 x 5,0 mm) y el PLCC 3528 (3,5 mm X 2,8 mm). La diferencia básica entre ambos es el tamaño del LED y la cantidad de luz que emite, siendo el más brillante el tipo 5050. Existen más tamaños de LED; pero su uso en cintas de LEDs actualmente es muy escaso.



Figura 40: Tipos de Led 2. [59]

CANTIDAD DE LEDS POR METRO

Este es un parámetro importante que nos indica la cantidad de LEDs que encontraremos en cada metro lineal de tira y, junto al tipo de LED, define cuanto va a iluminar cada segmento de la instalación. Las cantidades más habituales son 30, 60 y 90 LEDs por metro.

Todos estos datos se encuentran en la etiqueta de características que encontramos en el carrete o en el envoltorio de las tiras.



Figura 41: Cantidad de Led por metro. [59]

CONSUMO ELÉCTRICO

El consumo de energía de las tiras de LEDs depende de su potencia eléctrica. La potencia está dada por la cantidad y el tipo de LEDs que contenga cada metro de tira y, además de la potencia propia de los LEDs, también incluye el consumo propio de la circuitería auxiliar (disipación en las resistencias).

En la siguiente tabla se muestra el consumo aproximado de cada tipo de tira. [59]

Tipo de LED	LEDs/metro	Potencia Watt / m	Corriente A / m
3528	30	2,4	0,2
3528	60	4,8	0,4
3528	96	7,68	0,64
3528	120	9,6	0,8
3528	240	19,2	1,6
5050	30	7,2	0,6
5050	60	14,4	1,2
5050	120	28,8	2,4

www.afinidadeletrica.com

Figura 42: Consumo Eléctrico de Led. [59]

CANTIDAD DE LUZ QUE EMITE

Cada LED debidamente conectado y polarizado emite un flujo luminoso (medida de la potencia luminosa percibida) que depende de varios factores constructivos. Se pueden considerar los siguientes valores medios para los formatos de LEDs más usuales:

- SMD 3528 (3,5mm x 2,8mm): Cada LED emite entre 3 y 5 Lúmenes.
- SMD 5050 (5,0mm x 5,0mm): Cada LED emite entre 10 y 15 Lúmenes.

Por lo que para calcular el flujo luminoso total del segmento, simplemente se debe multiplicar la cantidad de LEDs que contiene por el flujo individual de cada pieza. [59]

Tipo de LED	LEDs/metro	Flujo luminoso lm / LED	Flujo luminoso lm / metro
3528	30	3 a 5	90 a 150
3528	60	3 a 5	180 a 300
3528	96	3 a 5	288 a 480
3528	120	3 a 5	360 a 600
3528	240	3 a 5	720 a 1200
5050	30	10 a 15	300 a 450
5050	60	10 a 15	600 a 900
5050	120	10 a 15	1200 a 1800

www.afinidadelectrica.com

Figura 43: Cantidad de Luz emitente. [59]

COLOR

Existen dos grandes clasificaciones de cintas de LEDs: Las monocolor y las RGB. Las primeras vienen de un solo color fijo y solo podrá variarse su intensidad con un atenuador o dimmer. Los colores más comunes son: blanco frío, blanco cálido, rojo, azul, verde y amarillo. Existen en el mercado colores especiales y tiras fijas con combinaciones de LEDs de diferentes colores. Las tiras monocolor se reconocen porque tienen solo dos terminales para su conexión: positivo y negativo. [59]



Figura 44: Tira Led Monocolor. [59]

Las tiras tipo RGB permiten seleccionar cada color, intensidad y modo de transición entre colores y para su comando se utiliza generalmente un controlador electrónico.

Las tiras RGB están compuestas por LEDs especiales llamados RGB que tienen en el mismo substrato tres LEDs, cada uno de un color básico (Red, Green, Blue).



Figura 45: Tira Led RGB. [59]

Para la selección del color luz se utiliza un criterio de síntesis aditiva del color. La síntesis aditiva es un método para obtener un color de luz determinado por la suma de otros colores. El proceso de reproducción aditiva normalmente usa luz roja, verde y azul para producir el resto de colores. Combinando uno de estos colores primarios con otro en proporciones iguales se producen los colores aditivos secundarios, más claros que los anteriores: cian, magenta y amarillo. Al variar la intensidad de cada color se revela el espectro completo de estas tres luces. La ausencia de los tres da el negro, y la suma de los tres da el blanco. Estos tres colores se corresponden con los tres picos de sensibilidad de los tres sensores de color en nuestros ojos. [59]

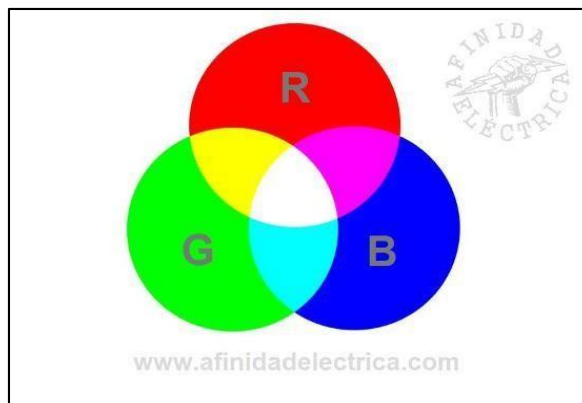


Figura 46: RGB. [59]

Las tiras RGB se identifican por contar con cuatro terminales para su conexión: Terminal positivo (+V), terminal rojo (R), terminal verde (G) y terminal azul (B).



Figura 47: Tira led RGB. [59]

GRADO DE PROTECCIÓN

Para situaciones en las que puedan quedar sometidas a daños mecánicos o a la acción de los elementos por estar instaladas a la intemperie, existen tiras con protección especial. Estas se clasifican con el grado de protección IP (Ingress Protection), según el estándar internacional IEC 60529 Degrees of Protection que indica el nivel de protección que sus materiales contenedores le proporcionan contra el ingreso de materiales extraños.

Teniendo en cuenta la anterior codificación, las tiras de LEDs pueden adquirirse con 3 niveles de protección diferentes:

- Sin protección (IP20 o IP33): No pueden mojarse ni instalarse en lugares húmedos, polvorientos o corrosivos. Sus ventajas son una mayor luminosidad y una mayor vida útil de los LEDs. Se utilizan para iluminación interior en habitaciones, oficinas de trabajo y comercios o como sustitutos de paneles con tubos fluorescentes.
- Recubiertas de epoxi (IP65): Tienen una capa de resina epoxi que recubre los LEDs dándoles una protección total contra el polvo además de una cierta protección al agua. Son adecuadas para la mayoría de los usos. No pueden mojarse ni instalarse en ambientes excesivamente húmedos o corrosivos, pero eventualmente pueden limpiarse con un paño humedecido en agua (sin productos de limpieza).
- Recubiertas en silicona (IP68): Estas tiras son sumergibles en agua de forma continuada por lo que son ideales para su instalación en piscinas, acuarios y fuentes. La desventaja del uso de la capa de silicona es que les resta a las tiras luminosidad y duración.

[59]



Figura 48: Grado de Protección Tira led RGB. [59]

VIDA ÚTIL

La vida útil de una fuente de luz es la cantidad de tiempo en la que ésta funciona sin perder mantenimiento lumínico, es decir sin disminuir su capacidad de iluminación o quemarse. En el caso de las tiras de LEDs, su vida útil estimada de 50000 horas. Este valor es muy superior al de la vida útil de otros sistemas de iluminación:

- Lámpara incandescente: 1000 horas.
- Lámparas halógenas: 2000 horas.
- Lámparas de bajo consumo CFL: 6000 horas.
- Tubos fluorescentes: 8000 horas. [59]

COMO ESTÁN COMPUESTAS LAS TIRAS DE LEDS

Como sabemos las tiras de LEDs son dispositivos electrónicos preensamblados, por lo que para su aplicación no es necesario conocer su circuito interno ni su funcionamiento. Las características de tensión de alimentación y potencia de la tira serán los únicos datos eléctricos necesarios para el cálculo de la instalación. De todas maneras, siempre es útil e ilustrativo comprender el funcionamiento interno de cualquier sistema por lo que a continuación se describe brevemente la topología interna de las tiras de LEDs monocolor y RGB. [59]

CIRCUITO INTERNO DE UNA TIRA DE LEDS MONOCOLOR

En el circuito interno de las tiras de LEDs monocolor encontramos sucesiones de grupos o arreglos de tres LEDs y una resistencia conectados en serie. Cada una de estas series se encuentra conectada en paralelo con la tensión de alimentación como se observa en el siguiente circuito:

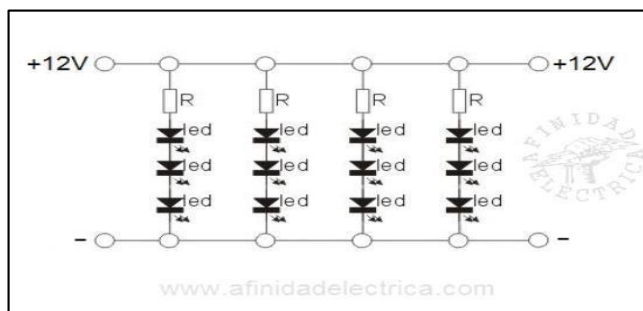


Figura 49: Circuito Internos de tira led Monocolor. [59]

CIRCUITO INTERNO DE UNA TIRA RGB

Los LEDs tipo RGB son chips que en su interior tienen tres LEDs (rojo, verde y azul) con terminales independientes. En tiras RGB encontramos los canales de distintos colores conectados en grupos de tres LEDs en serie con tres resistencias limitadoras (una para cada canal) y estos grupos, en paralelo con las líneas de alimentación.

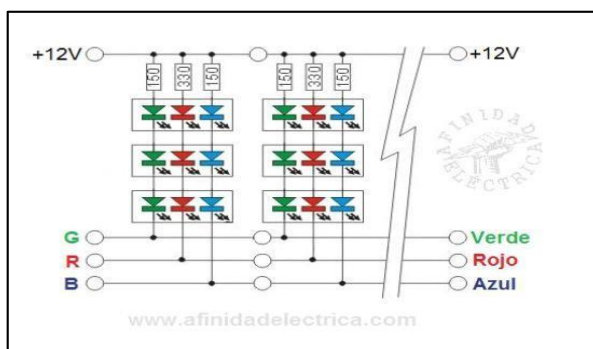


Figura 50: Circuito Internos de tira led RGB. [59]

Esta conexión permite cortar las tiras en cualquier punto cada tres LEDs. Se observa que el canal rojo utiliza una resistencia de mayor valor porque la tensión característica de la juntura roja es menor respecto a la de los otros colores.

Los grupos de LEDs están unidos por el polo positivo de alimentación por lo que los circuitos de control encienden los distintos LEDs llevando a masa las líneas R, G o B según necesidad. [59]

FUENTES DE ALIMENTACIÓN

Como buenos diodos que son, los LEDs (Light Emitting Diode o diodo emisor de luz) funcionan con corriente continua por lo que necesitan una fuente de alimentación para ser conectados a la red de corriente alterna.

Las fuentes de alimentación se seleccionan por la tensión que deberá ser la misma que la de las tiras y por la corriente total que se calcula según el largo de las tiras con la siguiente fórmula: [59]

$$\text{Corriente [A]} = \text{Potencia total [W]} / \text{Tensión [V]}$$

Dónde:

$$\text{Potencia total [W]} = \text{Potencia por metro de tira [W/m]} \times \text{Longitud de la tira [m]}$$

4.2.2.2. DRIVER DE POTENCIA PARA TIRA LED RGB

El driver de potencia para la tira led RGB está basado en transistores MOSFET IRCZ44 y se muestra en la Figura 51.

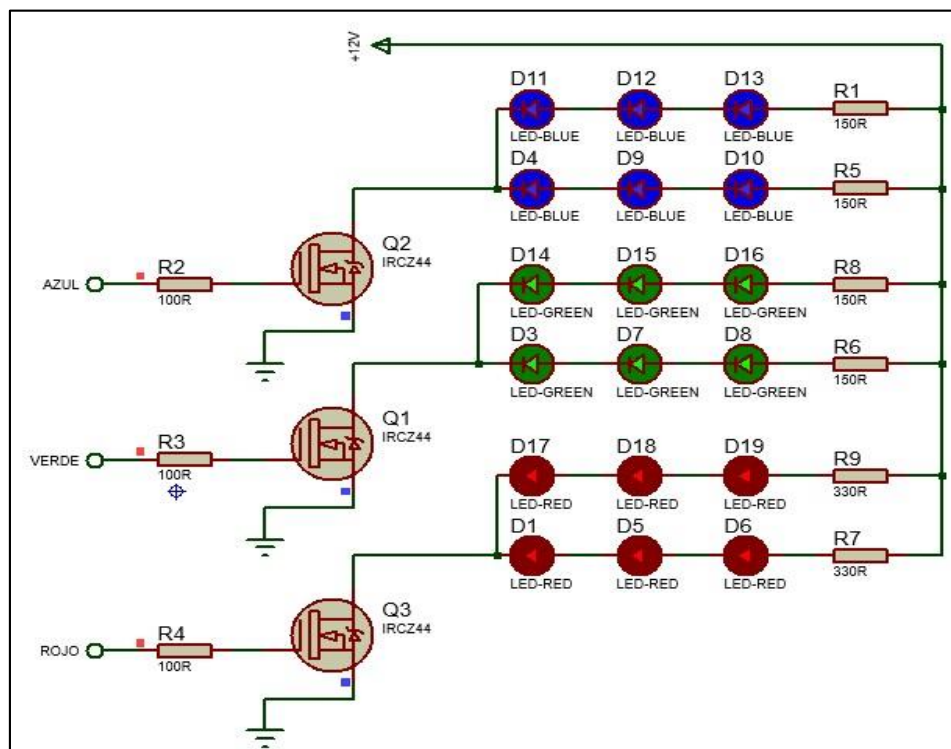


Figura 51: Driver de potencia para tira led RGB basado en el MOSFET IRCZ44. [PROPIO]

4.2.2.3. DRIVER DE POTENCIA PARA TIRA LED DE LAMPARA PARA FOTOTERAPIA

El driver de potencia para la tira led de la lámpara está basado en transistor MOSFET IRCZ44 y se muestra en la Figura 52.

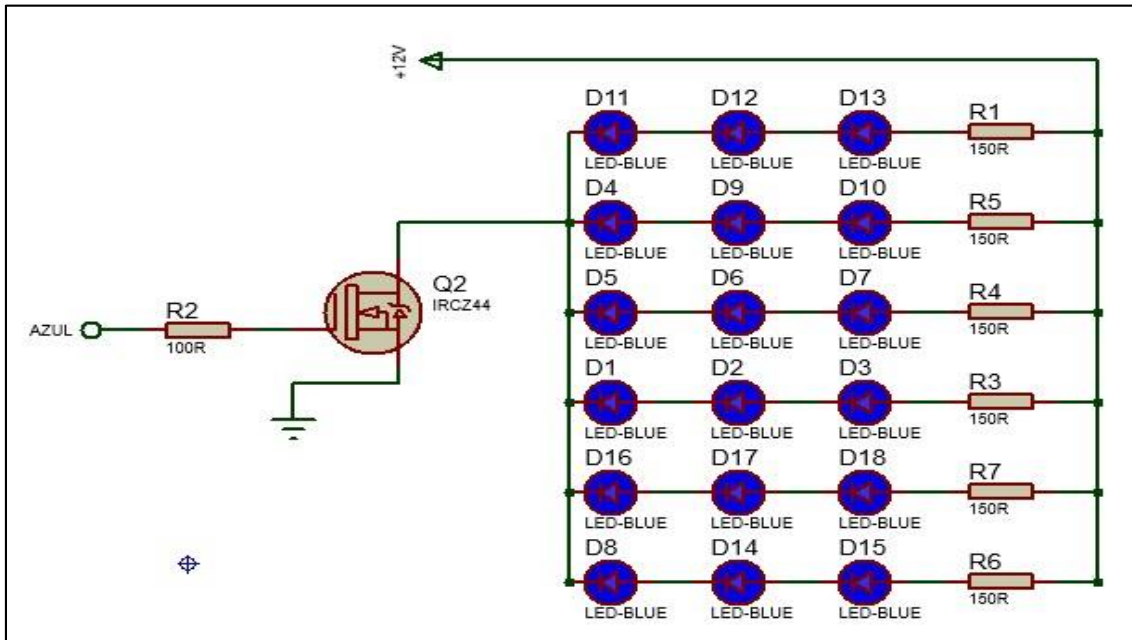


Figura 52: Driver de potencia para tira led para lámpara basado en el MOSFET IRCZ44. [PROPIO]

4.2.3. POSICIÓN DE LÁMPARA POR SERVOMOTOR

4.2.3.1. DEFINICIÓN DE SERVOMOTOR

Un servomotor (o servo) es un tipo especial de motor con características especiales de control de posición. Al hablar de un servomotor se hace referencia a un sistema compuesto por componentes electromecánicos y electrónicos. [60]

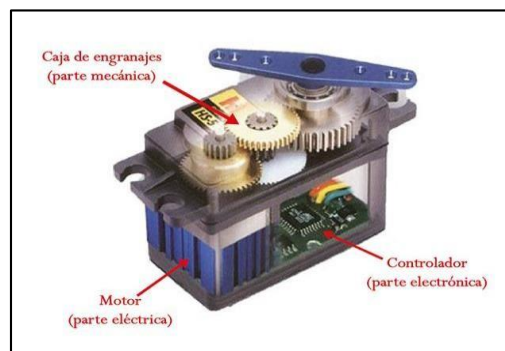


Figura 53: Servomotor. [60]

El motor en el interior de un servomotor es un motor DC común y corriente. El eje del motor se acopla a una caja de engranajes similar a una transmisión. Esto se hace para potenciar el torque del motor y permitir mantener una posición fija cuando se requiera. De forma similar a un automóvil, a menor mayor velocidad, menor torque. El circuito electrónico es el encargado de manejar el movimiento y la posición del motor. [60]

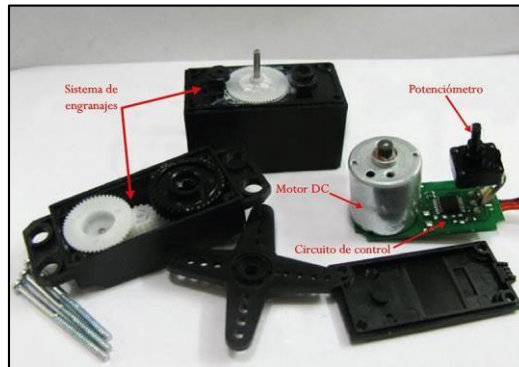


Figura 54: Partes de un Servomotor. [60]

La presencia del sistema de engranajes como el que se muestra en la figura hace que cuando movemos el eje motor se sienta una inercia muy superior a la de un motor común y corriente. Observando las imágenes que hemos presentado nos podemos dar cuenta que un servo no es un motor como tal, sino un conjunto de partes (incluyendo un motor) que forman un sistema. [60]

4.2.3.2. TIPOS DE SERVOMOTORES

Existen servomotores para todo tipo de usos. En la industria, la robótica, en el interior de las impresoras, máquinas CNC, etc. [60]

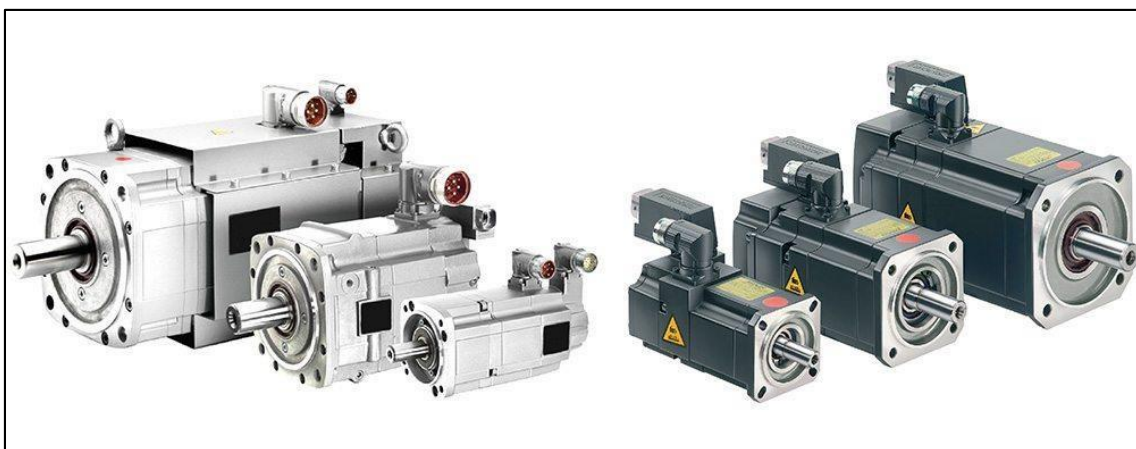


Figura 55: Servomotores Industriales. [60]

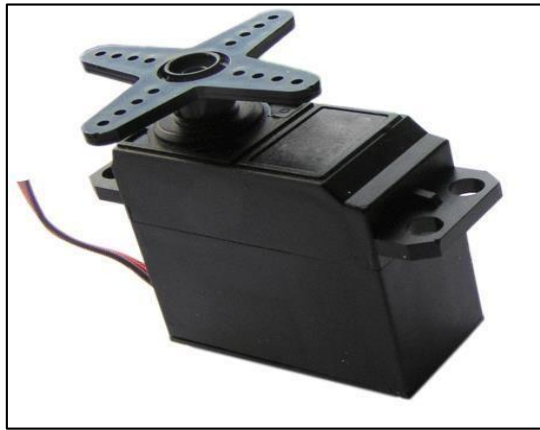


Figura 56: Servomotores para modelismo y Robótica. [60]

Los servomotores de modelismo operan a voltajes bajos en corriente directa, típicamente entre 4 y 6 voltios. Los servomotores industriales operan tanto en DC como en AC (monofásico o trifásico). Para mí es un poco difícil escribir sobre este tipo de temas debido a que nunca he tenido la oportunidad de trabajar con un servo industrial. Lo que sé sobre ellos es por lo que he leído o lo habré visto en algún video o seminario. Los servos de modelismo, en cambio, se pueden adquirir a muy bajo costo en Internet y son populares entre los usuarios de Arduino.

Se debe resaltar que, dentro de los diferentes tipos de servomotores, éstos se pueden clasificar según sus características de rotación. [60]

- **Servomotores de rango de giro limitado:** son el tipo más común de servomotor. Permiten una rotación de 180 grados, por lo cual son incapaces de completar una vuelta completa.
- **Servomotores de rotación continua:** se caracterizan por ser capaces de girar 360 grados, es decir, una rotación completa. Su funcionamiento es similar al de un motor convencional, pero con las características propias de un servo. Esto quiere decir que podemos controlar su posición y velocidad de giro en un momento dado. [60]

Los servomotores de rango de giro limitado se pueden adecuar para que funcionen como servomotores de rotación continua. Sin embargo, si requerimos un servo de 360 grados es mejor comprar uno que haya sido diseñado para este tipo de uso. [60]

4.2.3.3. FUNCIONAMIENTO DE UN SERVOMOTOR

Los servomotores poseen tres cables, a diferencia de los motores comunes que sólo tienen dos. Estos tres cables casi siempre tienen los mismos colores, por lo que son fácilmente reconocibles.

Voltaje positivo	Tierra (ground)	Señal de control
		

Figura 57: Colores comunes de los cables de un servomotor. [60]

Los colores dependerán del fabricante, pero difícilmente nos equivocaremos a la hora de reconocer los terminales de un servo. La necesidad de una señal de control para el funcionamiento de este tipo de motores hace que sea imposible utilizarlos sin un circuito de control adecuado. Esto se debe a que para que el circuito de control interno funcione, es necesaria una señal de control modulada. Para esto se utiliza modulación por ancho de pulsos, es decir, PWM. [60]

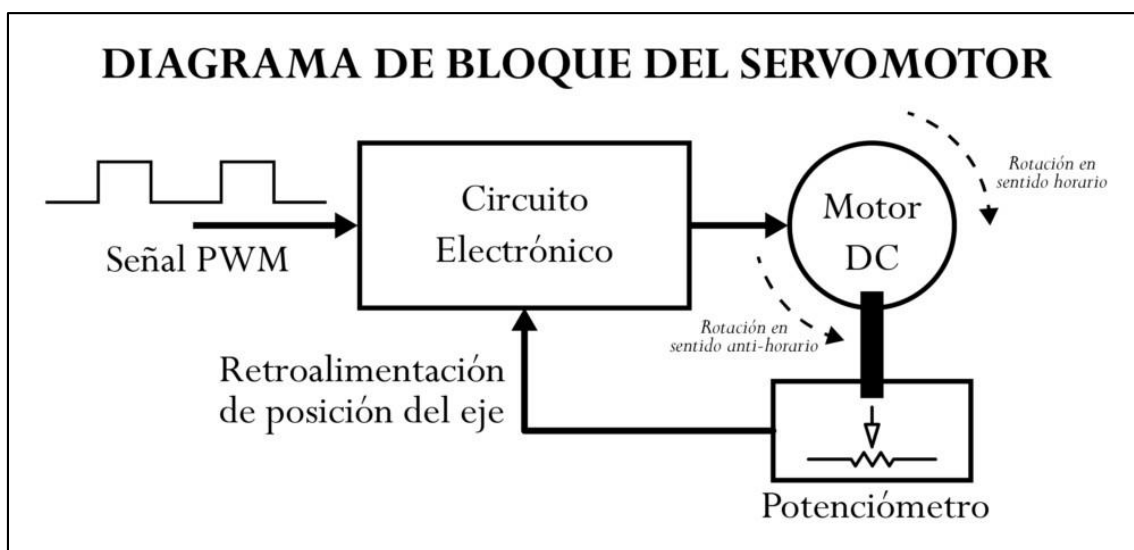


Figura 58: Diagrama de bloques del Servomotor. [60]

El diagrama de bloque del servomotor representa de forma visual el servomotor como un sistema. El circuito electrónico es el encargado de recibir la señal PWM y traducirla en movimiento del Motor DC. El eje del motor DC está acoplado a un potenciómetro, el cual permite formar un divisor de voltaje. El voltaje en la salida del divisor varía en función de la posición del eje del motor DC. [60]

En Panama Hitek ya hemos escrito sobre el funcionamiento de un potenciómetro. En dicho post presentamos una serie de gráficos sobre la variación del voltaje en el divisor de un potenciómetro en función del movimiento de la perilla. [60]

De forma similar, cuando el eje del motor modifica la posición del potenciómetro, el voltaje en la terminal central varía. El potenciómetro permite que el circuito de control electrónico pueda retroalimentarse con la posición del motor en un momento dado. Esto, en Teoría de Control se conoce como un sistema de lazo cerrado. [60]

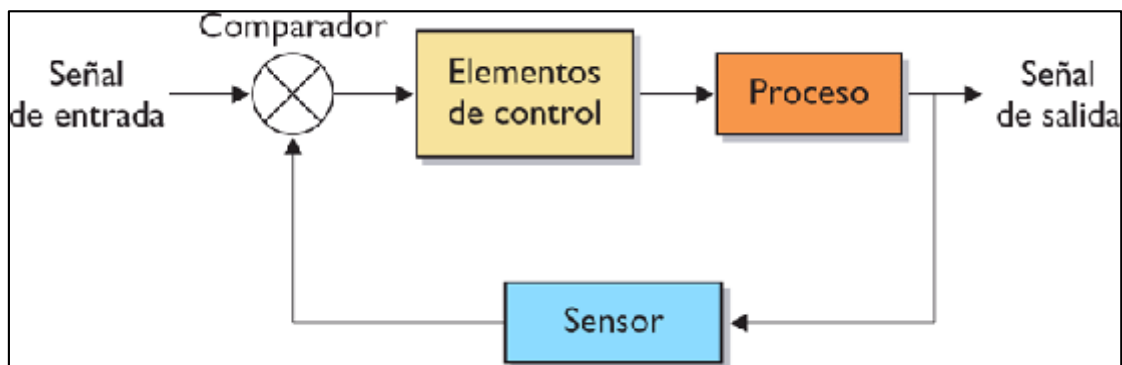


Figura 59: Representación en diagrama de bloques de un sistema de lazo cerrado. [60]

Los servomotores de rotación continua desacoplan el potenciómetro del eje del motor. Esto impide que el circuito de control pueda leer la posición del eje, por lo cual provoca un movimiento continuo al no ser capaz de cumplir la condición para que el servo se detenga. Los servomotores de rotación continua normalmente pueden girar en un sentido o en otro y detenerse. Podemos modificar la velocidad de giro, pero no podremos lograr, por ejemplo, que el servo se mueva una determinada cantidad de grados y luego se detenga. [60]

Las señales de PWM requeridas para que funcione el circuito de control electrónico son similares para la mayoría de los modelos de servo. Esta señal tiene la forma de una onda cuadrada. Dependiendo del ancho del pulso, el motor adoptará una posición fija. [60]

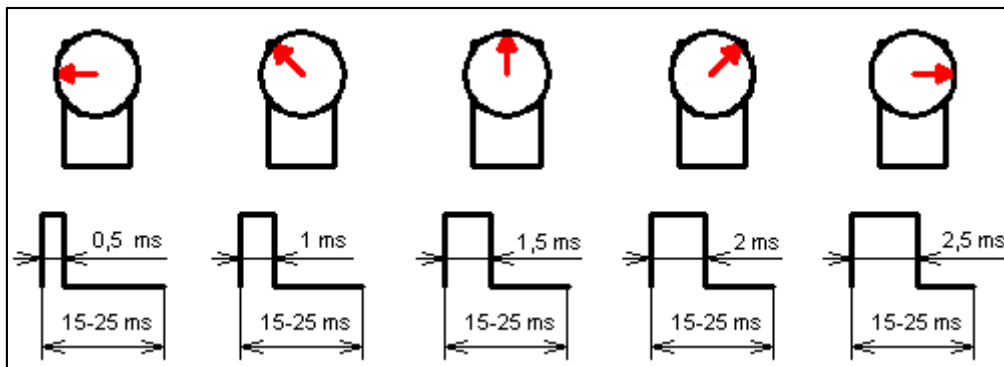


Figura 60: Señales de control de un Servomotor. [60]

Las señales que vemos en la imagen son las que permiten que el eje del motor adquiera determinada posición. Las señales deben repetirse en el tiempo para que el motor mantenga una posición fija. [60]

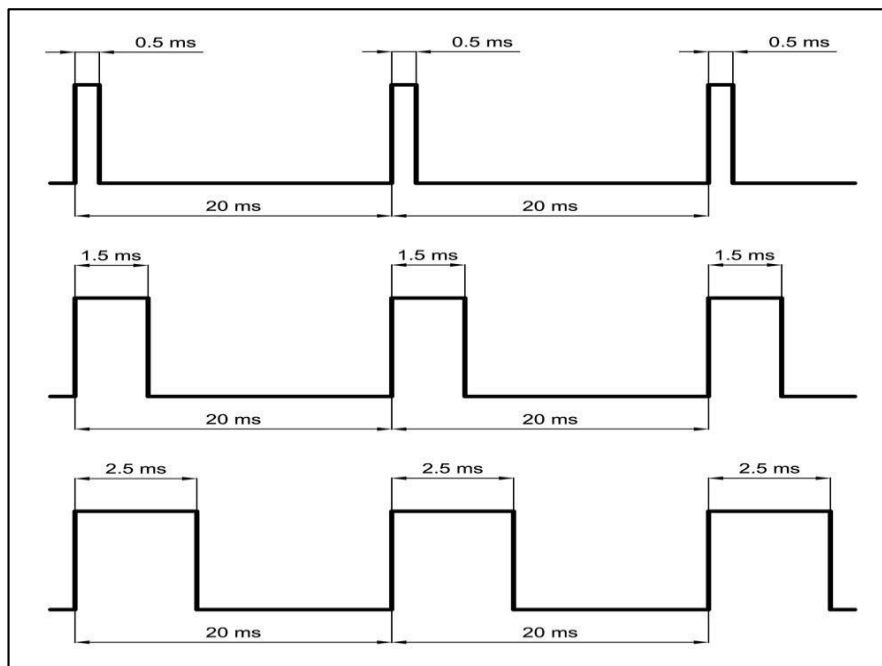


Figura 61: Ejemplos de trenes de pulsos para las posiciones 180°, 90° y 0° en el eje de un servomotor. [60]

La duración del ciclo de trabajo varía entre 15 y 25 milisegundos. Las ondas mostradas en la imagen anterior representan ejemplos de trenes de pulsos con los que se puede mover un servomotor, utilizando un ciclo de trabajo de 20 milisegundos.

Este tren de pulsos puede ser generado por un circuito oscilador (como un 555) o por un microcontrolador. Es decir, con Arduino podemos controlar fácilmente un servomotor. De hecho en Arduino existen las librerías para el control de servos de forma nativa. No es necesario descargarlas. [60]

4.2.3.4. CONSUMO DE ENERGÍA

La energía consumida por una carga eléctrica será igual a la potencia (producto del voltaje por la corriente que entra a la carga) multiplicada por el tiempo de uso del motor. Debido a que los servomotores son alimentados entre 4 y 6 voltios, es posible asumir que el voltaje de alimentación es casi constante para la mayoría de los modelos. Lo que no es igual para uno u otro modelo de servomotor es el consumo de corriente. La corriente demandada por un servomotor depende de diferentes parámetros:

- Fabricante y modelo del servo
- La inercia acoplada al eje del motor. Esto incluye la inercia del sistema de engranajes y la de la carga que esté acoplada al motor.
- La velocidad de rotación aplicada al eje del servo

A mayor consumo de corriente, mayor demanda de potencia y por extensión, mayor consumo de energía. Esto es especialmente importante a la hora de controlar este tipo de motores con un microcontrolador. El consumo de corriente de un servo casi siempre es superior a la capacidad máxima del microcontrolador, por lo que se recomienda utilizar fuentes externas.

Los microcontroladores no han sido diseñados para entregar corriente a una carga sino para entregar señales de control. Estas señales de control se utilizan para activar o desactivar transistores (BJT o MOSFET), los cuales han sido especialmente diseñados para manejar grandes corrientes o voltajes. [60]

4.2.3.5. SERVOMOTOR SELECCIONADO

El servomotor seleccionado es el que tiene las siguientes características:

- Engranajes de Nylon
- Voltaje de operación 4.8-6 Volts.
- Peso 38 gr.
- Velocidad de trabajo 0.23 seg/60 grados (4.8Volts), 0.19 seg/60 grados (6 Volts)
- Torque 3.2 kg/cm (4.8 volts), 4.1 kg/cm (6 Volts)
- Frecuencia de funcionamiento 50Hz

Los circuitos de conexión con el microcontrolador es el que se muestra en las Figuras 62 y 63.

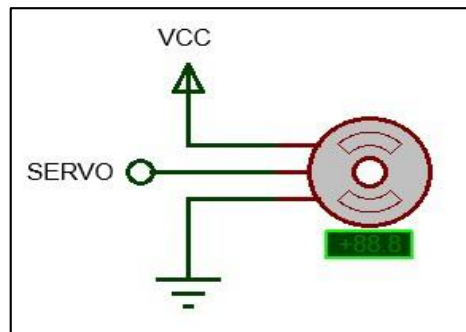


Figura 62: Servomotor de PROTEUS. [PROPIO]

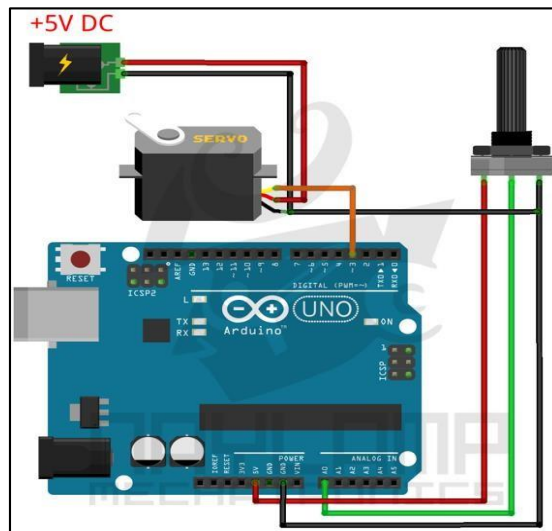


Figura 63: Conexión del servomotor con ARDUINO UNO. [PROPIO]

4.2.4. SENSORES DE DISTANCIA

Para medir distancias se utiliza el sensor HC-SR04.

4.2.4.1. DESCRIPCIÓN DEL FUNCIONAMIENTO DEL SENSOR HC-SR04

Es un sensor de distancias por ultrasonidos desarrollado por la firma DEVANTECH Ltda. Capaz de detectar objetos y calcular la distancia a la que se encuentra en un rango de 1,7 a 431 cm. (<http://www.micropik.com/>)

El sensor funciona por ultrasonidos y contiene toda la electrónica encargada de hacer la medición. Su uso es tan sencillo como enviar el pulso de arranque y medir la anchura del pulso de retorno. De muy pequeño tamaño, el sensor se destaca por su bajo consumo, gran precisión y bajo precio. [61]

El sensor funciona emitiendo impulsos de ultrasonidos inaudibles para el oído humano. Los impulsos emitidos viajan a la velocidad del sonido hasta alcanzar un objeto, entonces el sonido es reflejado y captado de nuevo por el receptor de ultrasonidos. Lo que hace el controlador incorporado es emitir una ráfaga de impulsos y a continuación empieza a contar el tiempo que tarda en llegar el eco. Este tiempo se traduce en un pulso de eco de anchura proporcional a la distancia a la que se encuentra el objeto. Registrando la duración del pulso es posible calcular la distancia en pulgadas, centímetros o en cualquier otra unidad de medida. Si no se detecta nada, entonces el HC-SR04 baja el nivel lógico de su línea de eco después de 30mS. [61]

El HC-SR04 proporciona un pulso de eco proporcional a la distancia. Si el ancho del pulso se mide en μS , el resultado se debe dividir entre 58 para saber el equivalente en centímetros, y entre 148 para saber el equivalente en pulgadas. $\mu\text{S}/58=\text{cm}$ o $\mu\text{S}/148=\text{pulgadas}$. El HC-SR04 puede activarse cada 50mS, o 20 veces por segundo. Debería esperar 50ms antes de la siguiente activación, incluso si el HC-SR04 detecta un objeto cerca y el pulso del eco es más corto. De esta manera se asegura que el "bip" ultrasónico ha desaparecido completamente y no provocará un falso eco en la siguiente medición de distancia. [61]

Desde un punto de vista práctico, lo que hay que hacer es mandar una señal de arranque en el pin 3 del HC-R04 y después leer el ancho del impulso que proporciona en el pin 2. Externamente se aplica, por parte del usuario, un pulso de disparo o trigger de 10 μS de duración mínima. Se inicia la secuencia. El módulo transmite un tren de pulsos o "burst" de 8 ciclos a 40KHz. En ese momento la señal de salida ECO pasa a nivel "1". Cuando la cápsula receptora recibe la señal transmitida como consecuencia de haber rebotado en un objeto (eco), esta salida pasa de nuevo a nivel "0". El usuario debe medir la duración del pulso de esta señal, es decir, el tiempo en que la señal eco se mantiene a "1". Con objeto de que el módulo se estabilice, se debe dejar un lapsus de tiempo de unos 20ms mínimo entre el momento en que la señal de eco pasa a "0" y un nuevo pulso de disparo que inicie el siguiente ciclo o medida. Esto permite realizar medidas cada 50ms o lo que es igual a 20 medidas por segundo. La duración del pulso eco de salida varía entre 100 μs y 25ms, en función de la distancia entre las cápsulas del módulo y el objeto. La velocidad del sonido es de 29,15 $\mu\text{s}/\text{cm}$ que, como realiza un recorrido de ida y vuelta, queda establecida en 58,30 $\mu\text{s}/\text{cm}$. Así pues el rango mínimo que se puede medir es de 1,7 cm (100 $\mu\text{s}/58$) y el máximo de 431 cm (25ms/58). En la Figura 64 se muestra diagrama de tiempos. [61]

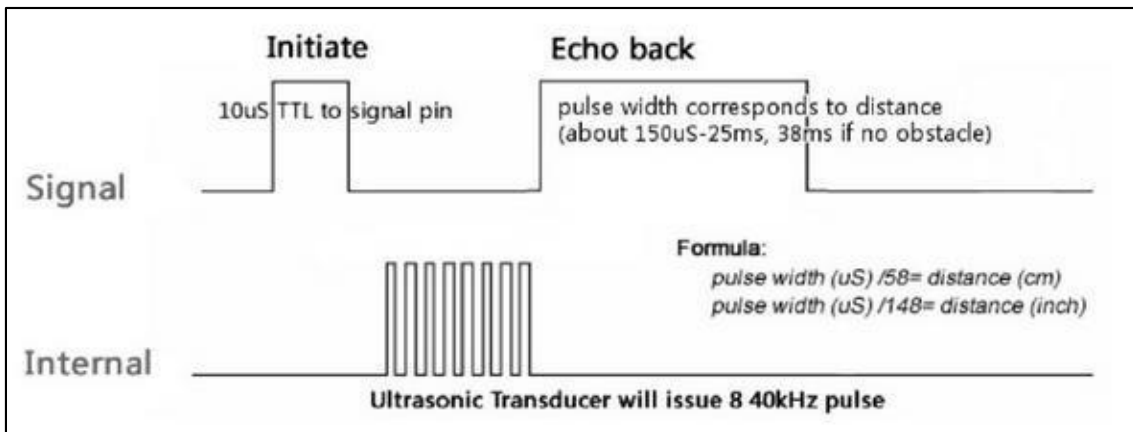


Figura 64: Diagrama de tiempos del sensor hc-sr04. [61]

Los circuitos de conexión con el microcontrolador es el que se muestra en la Figura 65.

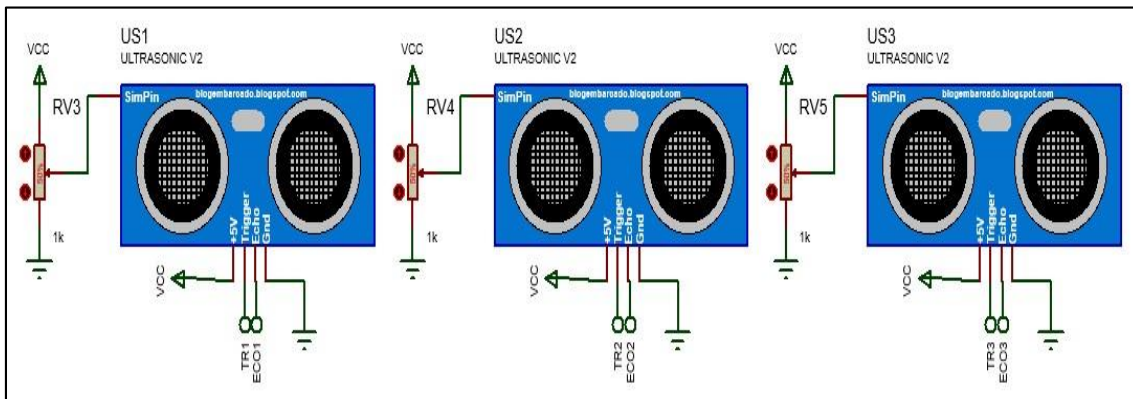


Figura 65: Diagrama de conexión del sensor hc-sr04 con ARDUINO. [PROPIO]

4.2.5. PANEL DE CONTROL

La lámpara de fototerapia puede ser controlada mediante comandos enviados inalámbricamente usando BLUETOOTH para lo cual se utiliza el módulo HC06, o mediante el uso de dos potenciómetros para control de potencia de los led y control de posición del servomotor.

Los circuitos de conexión con el microcontrolador es el que se muestra en la Figura 66.

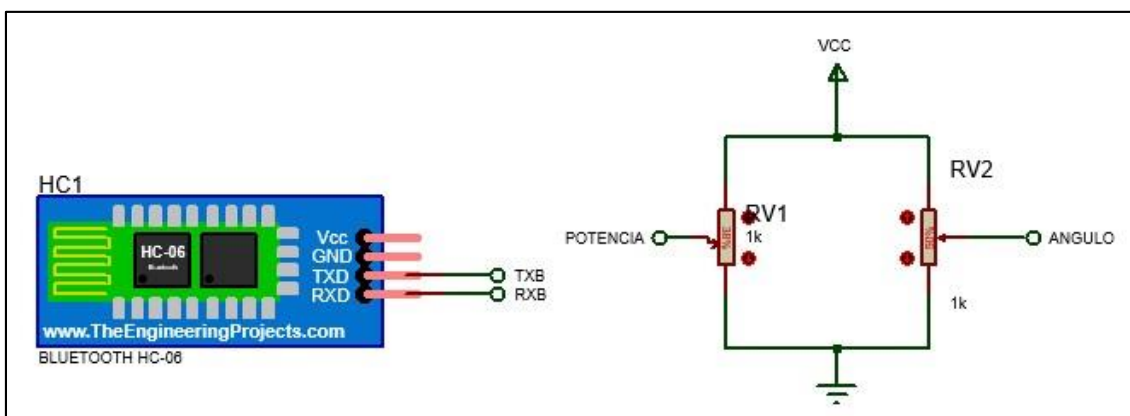


Figura 66: Diagrama de conexión del HC06 y potenciómetros de control de potencia y posición. [PROPIO]

4.2.6. DISPLAY LCD

Para la visualización del valor de los sensores de distancia, potencia entregada en los leds se utilizara una pantalla de cristal líquido LCD de 16x2.

Los circuitos de conexión con el microcontrolador es el que se muestra en la Figura 67.

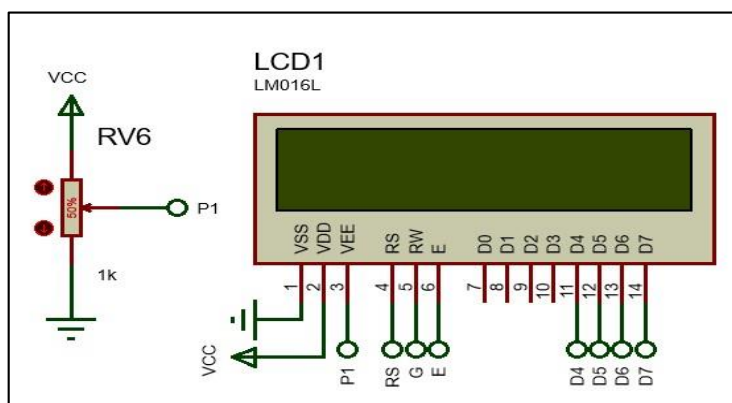


Figura 67: Diagrama de conexión de la pantalla LCD. [PROPIO]

4.2.7. MICROCONTROLADOR – ARDUINO UNO

Para la realización del proyecto se utilizó la tarjeta ARDUINO UNO el cual está basado en el microcontrolador ATMEGA328 el cual se describe a continuación

Arduino es una herramienta para hacer que los ordenadores puedan sentir y controlar el mundo físico a través de tu ordenador personal. Es una plataforma de desarrollo de computación física (physical computing) de código abierto, basada en una placa con un sencillo microcontrolador y un entorno de desarrollo para crear software (programas) para la placa. Puedes usar Arduino para crear objetos interactivos, leyendo datos de una gran variedad de interruptores y sensores y controlar multitud de tipos de luces, motores y otros actuadores físicos. Los proyectos con Arduino pueden ser autónomos o

comunicarse con un programa (software) que se ejecute en tu ordenador. La placa puedes montarla tú mismo o comprarla ya lista para usar, y el software de desarrollo es abierto y lo puedes descargar gratis desde la página www.arduino.cc/en/.

El Arduino puede ser alimentado a través de la conexión USB o con una fuente de alimentación externa. La fuente de alimentación se selecciona automáticamente.



Figura 68: ARDUINO UNO con cable de grabación [Propio]

4.2.7.1. ESPECIFICACIONES TECNICAS

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current for I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328)
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz

Tabla 12: Características del ARDUINO

4.2.7.2. POWER: PINES DE ALIMENTACION DEL ARDUINO UNO.



Figura 69: ARDUINO UNO - se indica los pines de alimentación [Propio]

Bien alimentemos al arduino mediante la conexión USB o mediante una fuente externa (recomendada de 7-12V), vamos a tener unas salidas de tensión continua debido a unos reguladores de tensión y condensadores de estabilización. Estos pines son:

VIN: se trata de la fuente tensión de entrada que contendrá la tensión a la que estamos alimentando al Arduino mediante la fuente externa.

5V: fuente de tensión regulada de 5V, esta tensión puede venir ya sea de pin VIN a través de un regulador interno, o se suministra a través de USB o de otra fuente de 5V regulada.

3.3V: fuente de 3.3 voltios generados por el regulador interno con un consumo máximo de corriente de 50mA.

GND: pines de tierra.

4.2.7.3. ENTRADAS Y SALIDAS DIGITALES



Figura 70: ARDUINO UNO indicando las entradas y salidas digitales [Propio]

Cada uno de los 14 pines digitales se puede utilizar como una entrada o salida. Cada pin puede proporcionar o recibir un máximo de 40 mA y tiene una resistencia de pull-up

(desconectado por defecto) de 20 a 50 kOhm. Además, algunos pines tienen funciones especializadas como:

- Pin 0 (RX) y 1 (TX). Se utiliza para recibir (RX) y la transmisión (TX) de datos serie TTL.
- Pin 2 y 3. Interrupciones externas. Se trata de pines encargados de interrumpir el programa secuencial establecido por el usuario.
- Pin 3, 5, 6, 9, 10 y 11. PWM (modulación por ancho de pulso). Constituyen 8 bits de salida PWM con la función `analogWrite()`.
- Pin 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Estos pines son de apoyo a la comunicación SPI.
- Pin 13. LED. Hay un LED conectado al pin digital 13. Cuando el pin es de alto valor, el LED está encendido, cuando el valor está bajo, es apagado.

4.2.7.4. ENTRADAS ANALÓGICAS

El Arduino posee 6 entradas analógicas, etiquetadas desde la A0 a A5, cada una de las cuales ofrecen 10 bits de resolución (es decir, 1024 estados). Por defecto, tenemos una tensión de 5V, pero podemos cambiar este rango utilizando el pin de AREF y utilizando la función `analogReference()`, donde le introducimos una señal externa de continua que la utilizara como referencia.



Figura 71: ARDUINO UNO indicando Entradas Analógicas [Propio]

En la Figura 72 se visualiza los pines del ARDUINO UNO que son utilizados para la implementación del proyecto.

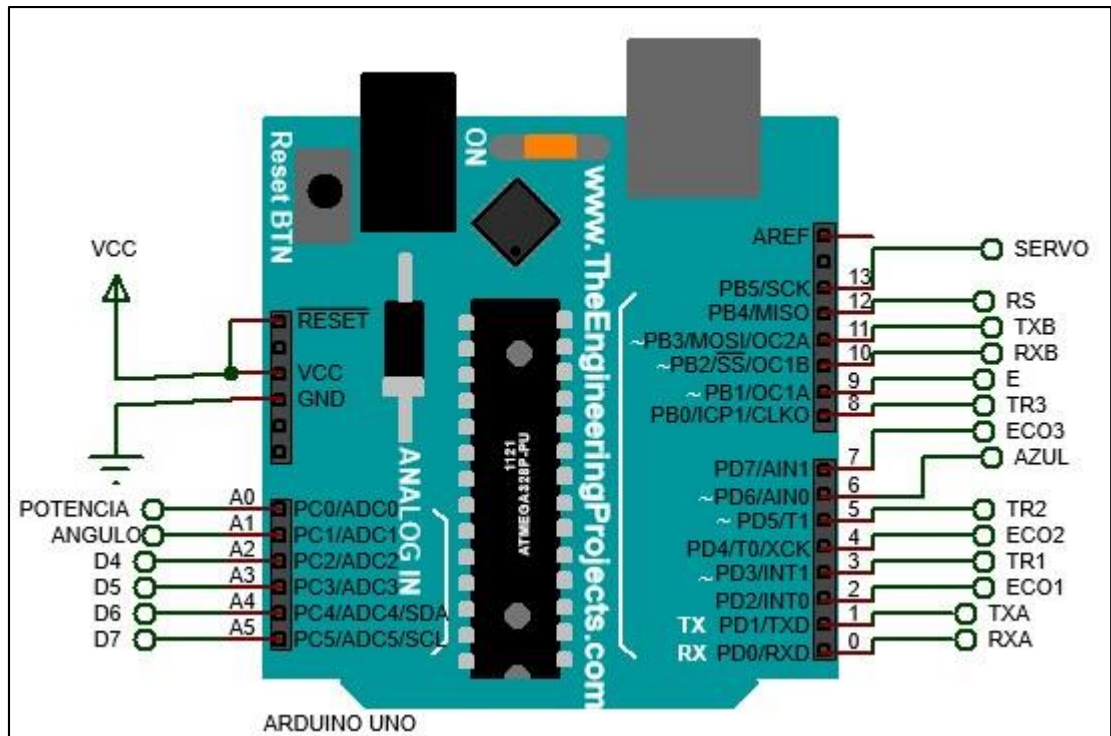


Figura 72: ARDUINO UNO indicando los pines utilizados para el proyecto [Propio]

4.3. DISEÑO DEL HARDWARE FINAL

De acuerdo a lo visto en párrafos anteriores e investigando cada una de las partes de nuestro sistemas, el diseño de la tarjeta electrónica para el sistema de control de la lámpara de Fototerapia se muestra en la Figura 73., en la Figura 74. Se muestra el mismo circuito para poder simular en PROTEUS.

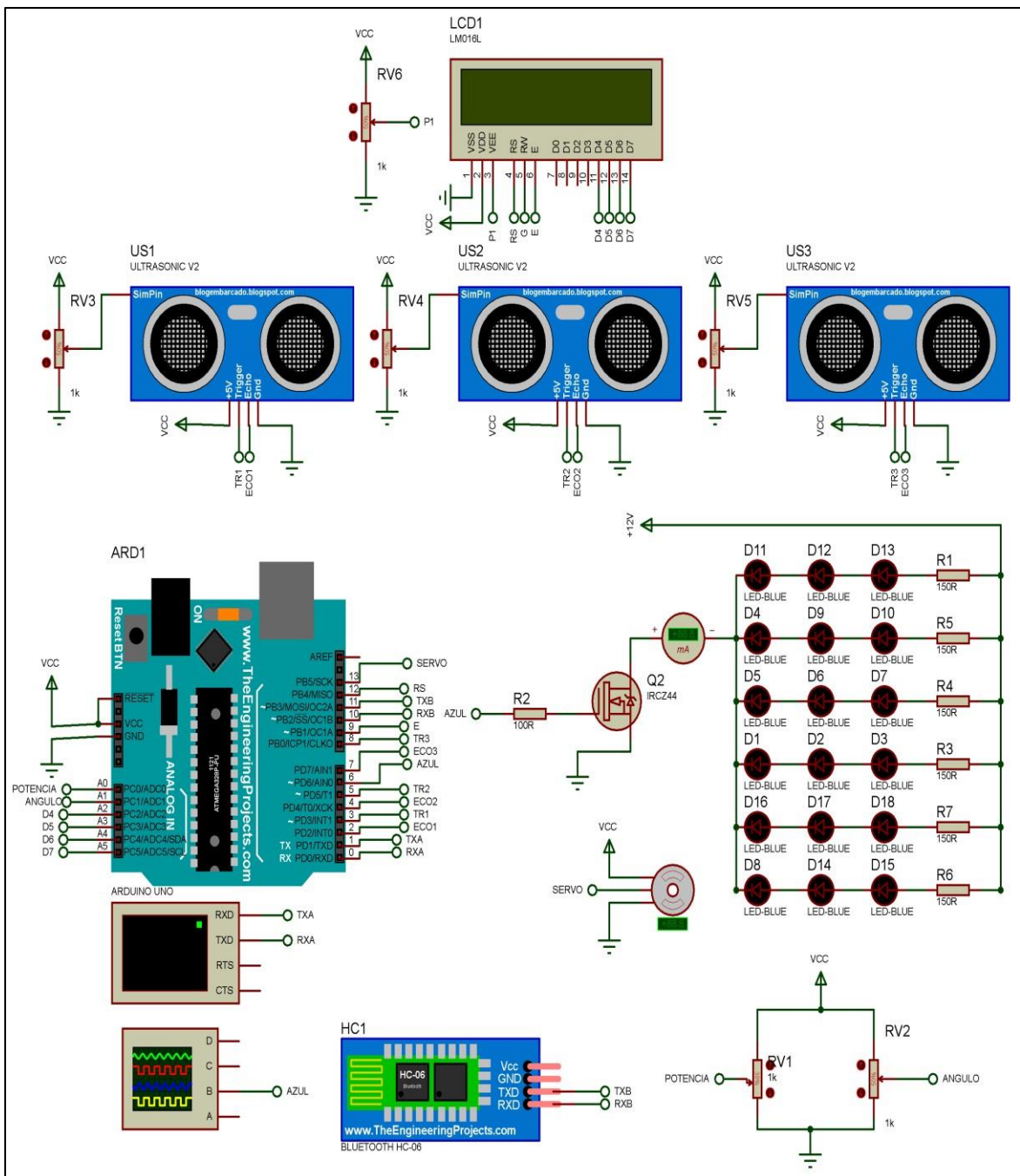


Figura 73: Diseño de tarjeta electrónica para sistema de control de lámpara de fototerapia led.
[Propio]

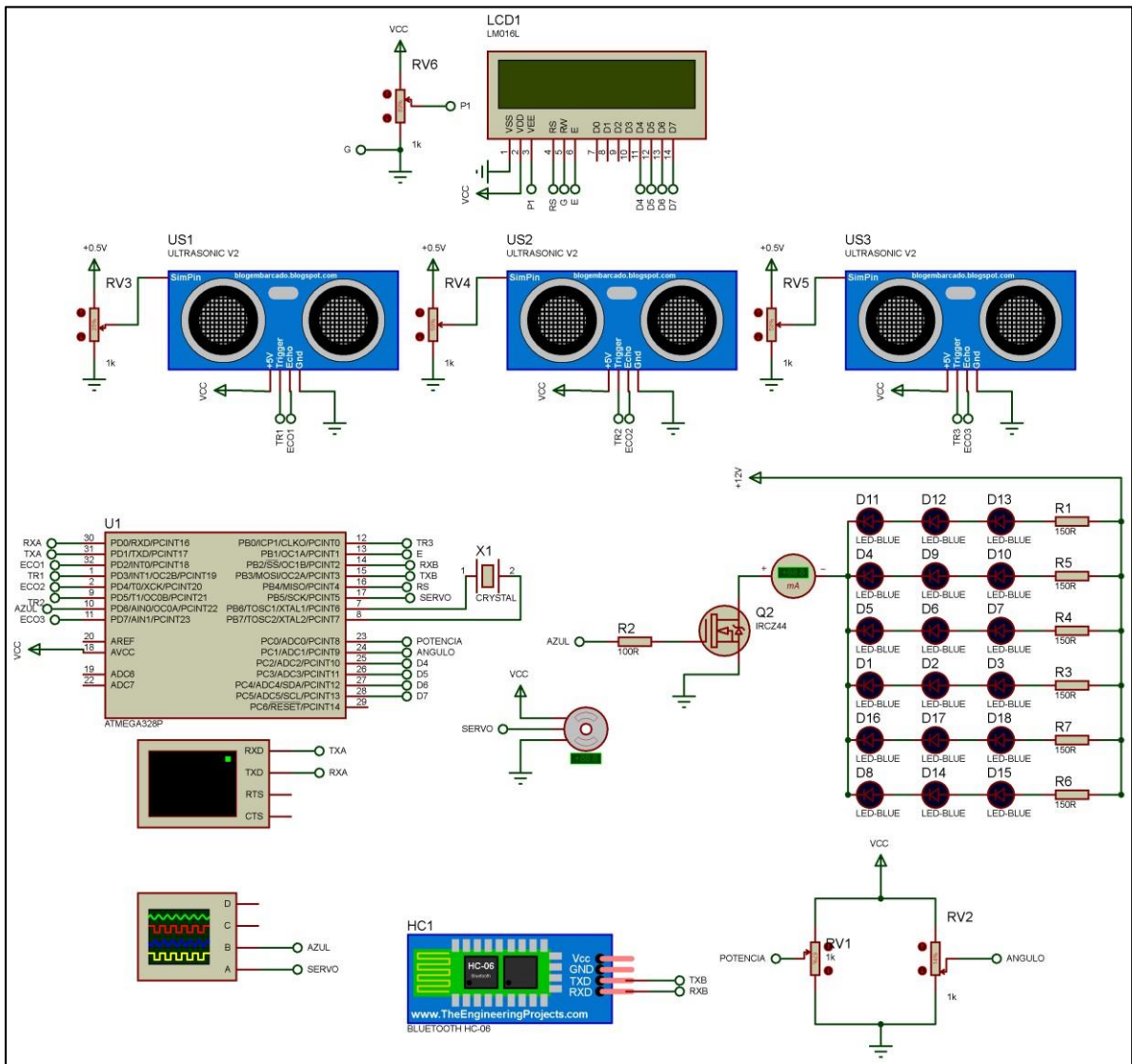


Figura 74: Diseño de tarjeta electrónica para sistema de control de lámpara de fototerapia led para simulación en PROTEUS. [Propio]

4.4. DISEÑO DEL SOFTWARE

El diseño del software se divide en las siguientes partes:

- Declaración de variables y configuraciones
- Comunicación Serial o Bluetooth
- Control de potencia de los leds
- Control de posición
- Medición de distancias
- Visualización de datos en pantalla LCD

A continuación se presentara los códigos de programa para ARDUINO UNO.


```

#define DEBUG_ARRAY(a) {for (int index = 0; index < sizeof(a) / sizeof(a[0]); index++)
#include <LiquidCrystal.h>
#include <Servo.h>
LiquidCrystal lcd(12, 9, A2,A3, A4,A5); // ( RS, EN, d4, d5, d6, d7)
Servo Motor; // Declaramos la variable para controlar el servo

// DECLARACIONES PARA SENSORES DE DISTANCIA
int trigPin=3; //Pin de trigger conectado en el pin 3 de la arduino
int echoPin=2; //Pin de ECHO conectado al 2 del arduino
int trigPin2=5; //Pin de trigger conectado en el pin 5 de la arduino
int echoPin2=4; //Pin de ECHO conectado al 4 del arduino
int trigPin3=8; //Pin de trigger conectado en el pin 8 de la arduino
int echoPin3=7; //Pin de ECHO conectado al 7 del arduino
float pingTime; // TIEMPO QUE SE DEMORA EN RETORNAR EL PULSO DEL SENSOR DE ULTRASONIDO
float targetDistance; //DISTANCIA EN PULGADAS
float speedOfSound=776.5; // VELOCIDAD DEL SONIDO EN MILLAS POR HORA.
float d_horizontal; // VALOR DE LA DISTANCIA HORIZONTAL TIPO FLOAT
float d_vertical1; // VALOR DE LA DISTANCIA VERTICAL 1 TIPO FLOAT
float d_vertical2; // VALOR DE LA DISTANCIA VERTICAL 2 TIPO FLOAT
int lcd_horizontal; // VALOR DE LA DISTANCIA HORIZONTAL TIPO INT
int lcd_vertical1; // VALOR DE LA DISTANCIA VERTICAL 1 TIPO INT
int lcd_vertical2; // VALOR DE LA DISTANCIA VERTICAL 2 TIPO INT

```

Figura 75: Código para definiciones y declaración de variables [PROPIO]

```

//DECLARACIONES PARA PONTENCIA DE LEDS Y ANGULO DEL SERVOMOTOR
const int pot_potencia = A0; // POTENCIOMETRO PARA POTENCIA EN CANAL A0
const int pot_angulo = A1; // POTENCIOMETRO PARA ANGULO EN CANAL A1
int valor_potencia; // VALOR DE LA POTENCIA TIPO INT
float valor_angulo; // VALOR DE LA VARIABLE valor_angulo TIPO FLOAT
float angulo; //VALOR DE LA VARIABLE angulo TIPO FLOAT
int angulo1; // VARIABLE ANGULO1 TIPO INT VALOR FINAL PARA EL SERVOMOTOR
int modo; // VARIABLE MODO INDICA SI SE VA CONTROLAR POR PUERTO SERIAL/BLUETOOTH
int azul=6; // PUERTO 6 PARA CONTROL PWM PARA TIRA LED

// DECLARACION PARA RECEPCION DE COMANDOS POR PUERTO SERIAL /BLUETOOTH
String str = "";
const char separator = ',';
const int dataLength = 3;
int data[dataLength];
//*****

```

Figura 76: Código para definiciones y declaración de variables [PROPIO]

```

//*****
void setup()
{
    Serial.begin(9600);
    pinMode(azul,OUTPUT); //CONFIGURANDO COMO SALIDA EL PUERTO 6 "AZUL"
    Serial.println("LAMPARA DE FOTOTERAPIA LED");
    lcd.begin(16, 2); // Fijar el numero de caracteres y de filas del LCD
    Motor.attach(13); // INDICA QUE EN EL PUERTO 13 DEL ARDUINO ESTA CONECTADO EL SERVOMOTOR
}
//*****

```

Figura 77: Código de configuraciones iniciales [PROPIO]

```

void loop()
{
    distancias();//EJECUTA RUTINA DE LEER SENSORES DE DISTANCIA (ULTRASONIDO)
    ver_lcd();//EJECUTA RUTINA PARA VISUALIZAR DATOS DE DISTANCIA, POTENCIA Y ANGULO DE SERVOMOTO EN LCD

    //*****
    //LEENDO PUERTO SERIAL A LA ESPERA DE COMANDOS
    if (Serial.available())// SI PUERTO SERIAL ESTA DISPONIBLE
    {
        str = Serial.readStringUntil('\n'); // RECIBE CADENA HASTA QUE LLEGA EL ENTER
        for (int i = 0; i < dataLength ; i++)// LEE TRES DATOS
        {
            int index = str.indexOf(separator);// INDEX GUARDA POSICION HASTA ENCONTRAR SEPARADOR ","
            data[i] = str.substring(0, index).toInt();//SE ALMACENA DATO HASTA EL SEPARADOR
            str = str.substring(index + 1);// SE INCREMENTA LA SIGUIENTE POSICION DE LA CADENA DESPUES DEL SEPARADOR ","
        }
        DEBUG_ARRAY(data);// SE ENVIA POR EL PUERTO SERIAL LOS DATOS
    }
    //*****

```

Figura 78: Código de lazo principal de programa – lectura de puerto serial [PROPIO]

```

//*****MODO 0 Y MODO 1*****
modo=data[0];// SE DEFINE EL MODO SI ES 0 SE CONTROLA POR PUERTO SERIAL LA
// SI ES 1 SE CONTROLA POR POTENCIOMETRO
if(modo==0){ //MODO 0 CONTROL POR PUERTO SERIAL
    Motor.write(data[1]); //Angulo de 0 a 180 grados EN SERVOMOTOR
    analogWrite(azul,data[2]);// Se envia Potencia "PWM" de leds de lampara
    angulo1=data[1]; // se almacena el numero entero del angulo
    valor_potencia=data[2]);// Se almacena el valor de la potencia
}
if (modo==1){
    valor_potencia = analogRead(pot_potencia); // lee el valor del voltaje e
    valor_angulo = analogRead(pot_angulo); // lee el valor del voltaje entre
    valor_potencia =valor_potencia/4;// se acondiciona la variable para valo
    valor_angulo =valor_angulo/4;//se acondiciona la variable para valores 0
    angulo= (valor_angulo/255); // se acondiciona para valores de 0-180 grado
    angulo=angulo*180;// se acondiciona para valores de 0-180 grados
    angulo1=angulo;// se obtiene el numero entero de la variable angulo
    Motor.write(angulo1); //Se envia Angulo de 0 a 180 grados al servomotor
    analogWrite(azul,valor_potencia);//se envia Potencia de lampara (PWM)
}
//*****

```

Figura 79: Código de modos de trabajo por puerto serial o por potenciómetros [PROPIO]


```

//*****LEER DISTANCIAS DE 3 SENSORES EN cm*****
void distancias(){
  digitalWrite(trigPin, LOW); // ESTADO BAJO "0" EL PIN DE TRIGER
  delayMicroseconds(2000); // RETARDO DE 2000US
  digitalWrite(trigPin, HIGH); // ESTADO ALTO "1" EL PIN DE TRIGER
  delayMicroseconds(15); // RETARDO DE 15US
  digitalWrite(trigPin, LOW); // ESTADO BAJO "0" EL PIN DE TRIGER
  delayMicroseconds(10); // RETARDO DE 10US
  pingTime = pulseIn(echoPin, HIGH); //pingTime is presented in microseconds
  pingTime=pingTime/1000000; //convierte pingTime a segundos dividiendo entre 1000000 (microsegundos)
  pingTime=pingTime/3600; //convierte pingtime a horas dividiendo entre 3600 (segundos en horas)
  d_horizontal= speedOfSound * pingTime; //Obteniendo distancia ,esto será en millas, ya que la velocidad del sonido es en millas por hora
  d_horizontal=d_horizontal/2; //Recuerde que el ping(pulso) se desplaza hacia el objetivo ida y vuelta
  d_horizontal= d_horizontal*63360; //Convertir millas a pulgadas multiplicando por 63360
  d_horizontal= d_horizontal*2.54; // obtener centimetros las pulgadas se multiplica por 2.54
  lcd_horizontal=d_horizontal;// se obtiene el numero entero de la distancia em cm
}

```

Figura 80: Código para medición de distancia horizontal [PROPIO]

```

digitalWrite(trigPin2, LOW); // ESTADO BAJO "0" EL PIN DE TRIGER
delayMicroseconds(2000); // RETARDO DE 2000US
digitalWrite(trigPin2, HIGH); // ESTADO ALTO "1" EL PIN DE TRIGER
delayMicroseconds(15); // RETARDO DE 15US
digitalWrite(trigPin2, LOW); // ESTADO BAJO "0" EL PIN DE TRIGER
delayMicroseconds(10); // RETARDO DE 10US
pingTime = pulseIn(echoPin2, HIGH); //pingTime almacena el tiempo de estado 1, tiempo en llegar al objetivo
pingTime=pingTime/1000000; //convierte pingTime a segundos dividiendo entre 1000000 (microsegundos)
pingTime=pingTime/3600; //convierte pingtime a horas dividiendo entre 3600 (segundos en horas)
d_verticall= speedOfSound * pingTime; //Obteniendo distancia ,esto será en millas, ya que la velocidad del sonido es en millas por hora
d_verticall=d_verticall/2; //Recuerde que el ping(pulso) se desplaza hacia el objetivo ida y vuelta
d_verticall= d_verticall*63360; //Convertir millas a pulgadas multiplicando por 63360
d_verticall= d_verticall*2.54; // obtener centimetros las pulgadas se multiplica por 2.54
lcd_verticall=d_verticall;// se obtiene el numero entero de la distancia em cm

```

Figura 81: Código para medición de distancia vertical 1 [PROPIO]

```

digitalWrite(trigPin3, LOW); // ESTADO BAJO "0" EL PIN DE TRIGER
delayMicroseconds(2000); // RETARDO DE 2000US
digitalWrite(trigPin3, HIGH); // ESTADO ALTO "1" EL PIN DE TRIGER
delayMicroseconds(15); // RETARDO DE 15US
digitalWrite(trigPin3, LOW); // ESTADO BAJO "0" EL PIN DE TRIGER
delayMicroseconds(10); // RETARDO DE 10US
pingTime = pulseIn(echoPin3, HIGH); //pingTime almacena el tiempo de estado 1, tiempo en llegar al objetivo
pingTime=pingTime/1000000; //convierte pingTime a segundos dividiendo entre 1000000 (microsegundos)
pingTime=pingTime/3600; //convierte pingtime a horas dividiendo entre 3600 (segundos en horas)
d_vertical2= speedOfSound * pingTime; //Obteniendo distancia ,esto será en millas, ya que la velocidad del sonido es en millas por hora
d_vertical2=d_vertical2/2; //Recuerde que el ping(pulso) se desplaza hacia el objetivo ida y vuelta
d_vertical2= d_vertical2*63360; //Convertir millas a pulgadas multiplicando por 63360
d_vertical2= d_vertical2*2.54; // obtener centimetros las pulgadas se multiplica por 2.54
lcd_vertical2=d_vertical2; // se obtiene el numero entero de la distancia em cm

```

Figura 82: Código para medición de distancia vertical 2 [PROPIO]

```

//*****LCD 16X2*****
// VISUALIZAR EN EL LCD DE 16X2 LOS VALORES DE LAS VARIABLES DE DISTANCIA POTENCIA Y ANGULO
void ver_lcd(){
    lcd.clear();//limpiar lcd
    lcd.setCursor(0,0);// cursos columna 0 fila 0 (primera fila)
    lcd.print("H: "); // visualiza H:
    lcd.setCursor(2, 0); // cursor columna 2 primera fila
    lcd.print(lcd_horizontal) ;// visualiza valor de distancia horizontal
    lcd.setCursor(5,0);//cursor columna 5 primera fila
    lcd.print("V1: "); // visualiza V1:
    lcd.setCursor(8, 0); // cursor columna 8 primera fila
    lcd.print(lcd_vertical1) ;// visualiza distancia vertical 1
    lcd.setCursor(11,0); //cursor columna 11 primera fila
    lcd.print("V2:"); //visualiza V2:
    lcd.setCursor(14,0);// cursor columna 14 primera fila
    lcd.print(lcd_vertical2);// visualiza distancia vertical 2
    lcd.setCursor(0,1); //cursor columna 0 segunda fila
    lcd.print("POT:"); //visualiza POT:
    lcd.setCursor(4,1);// cursor columna 4 segunda fila
    lcd.print(valor_potencia); //visualiza valor de la potencia 0-255 en leds
    lcd.setCursor(9,1);// cursor columna 9 segunda fila
    lcd.print("ANG:"); //visualiza ANG:
    lcd.setCursor(13,1);// cursor columna 13 segunda fila
    lcd.print(angulo1); //visualiza valor del angulo para servomotor de 0-180 grados
}

```

Figura 83: Código para visualización de datos en pantalla LCD [PROPIO]

4.5. SIMULACIONES DEL DISEÑO DE LAMPARA DE FOTOTERAPIA

En las siguientes figuras se visualizaran pantallas de las simulaciones realizadas:

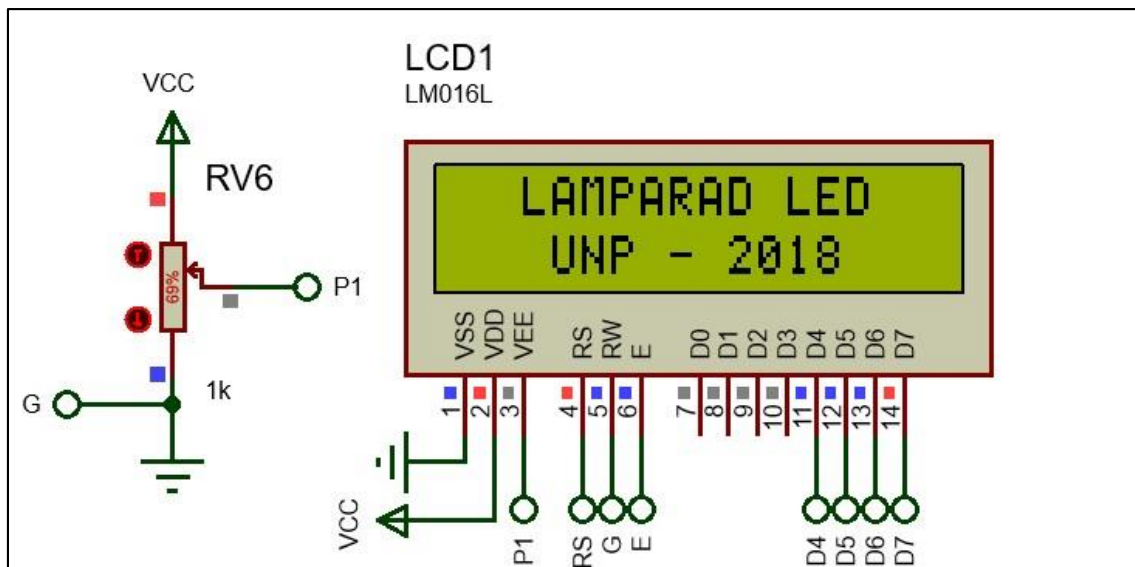


Figura 84: Mensaje en LCD al Iniciar Programa [PROPIO]

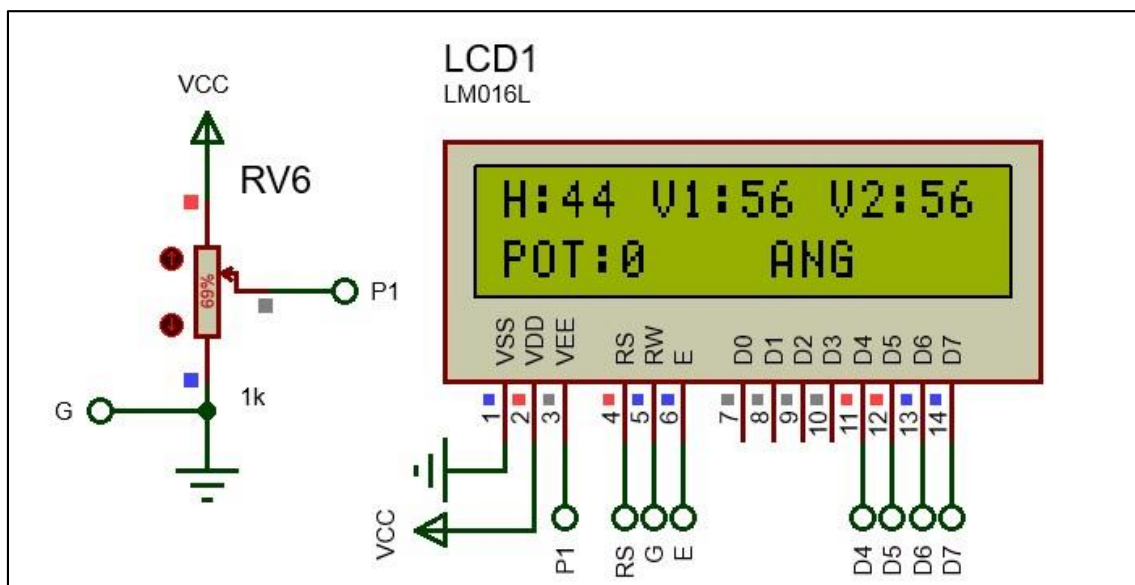


Figura 85: Mensaje en LCD al Iniciar Programa visualiza distancias [PROPIO]

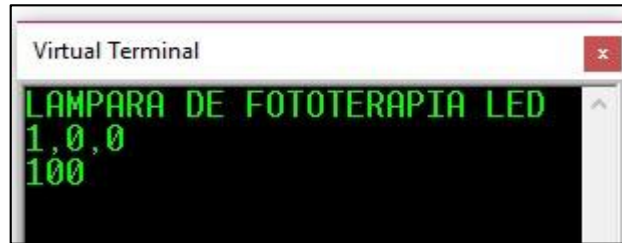


Figura 86: Configurando por Puerto Serial para controlar por potenciómetros [PROPIO]

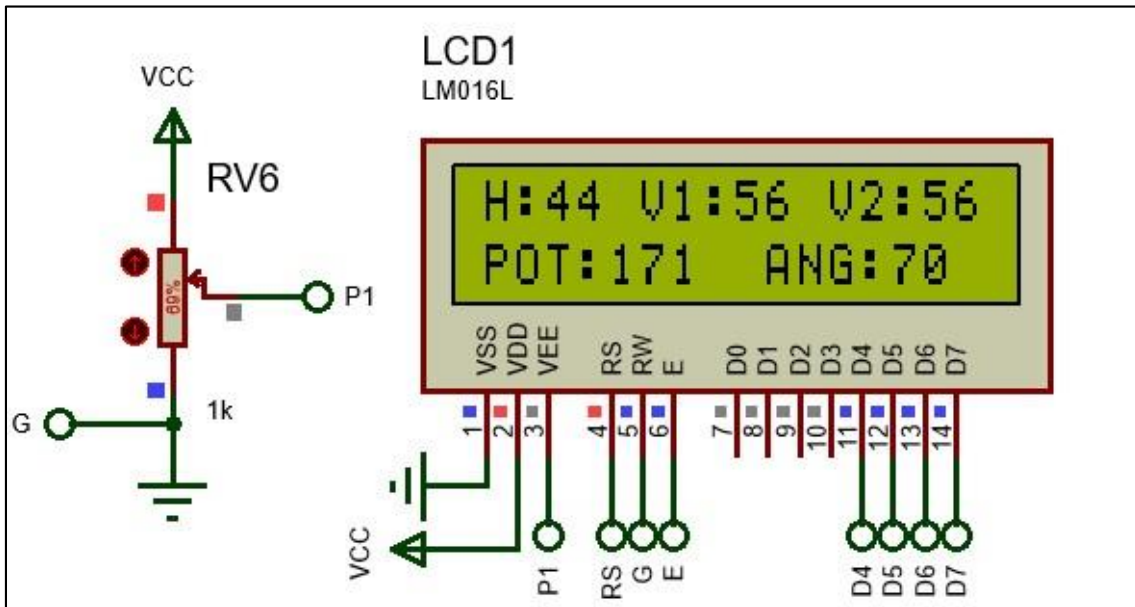


Figura 87: Mensaje en LCD visualizando distancias, Potencias y ángulo del servomotor controlado por potenciómetros. [PROPIO]

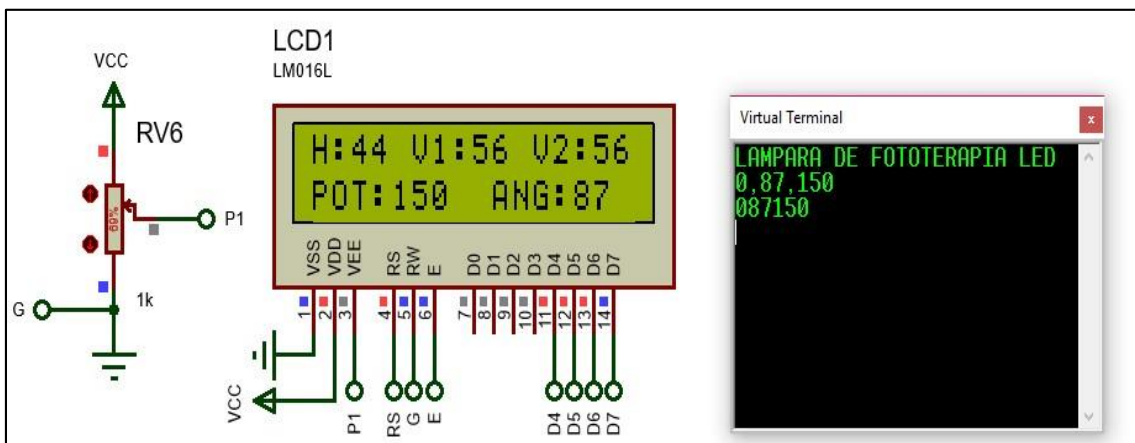


Figura 88: Mensaje en LCD visualizando distancias, Potencias y ángulo del servomotor controlado por puerto serial USB. [PROPIO]

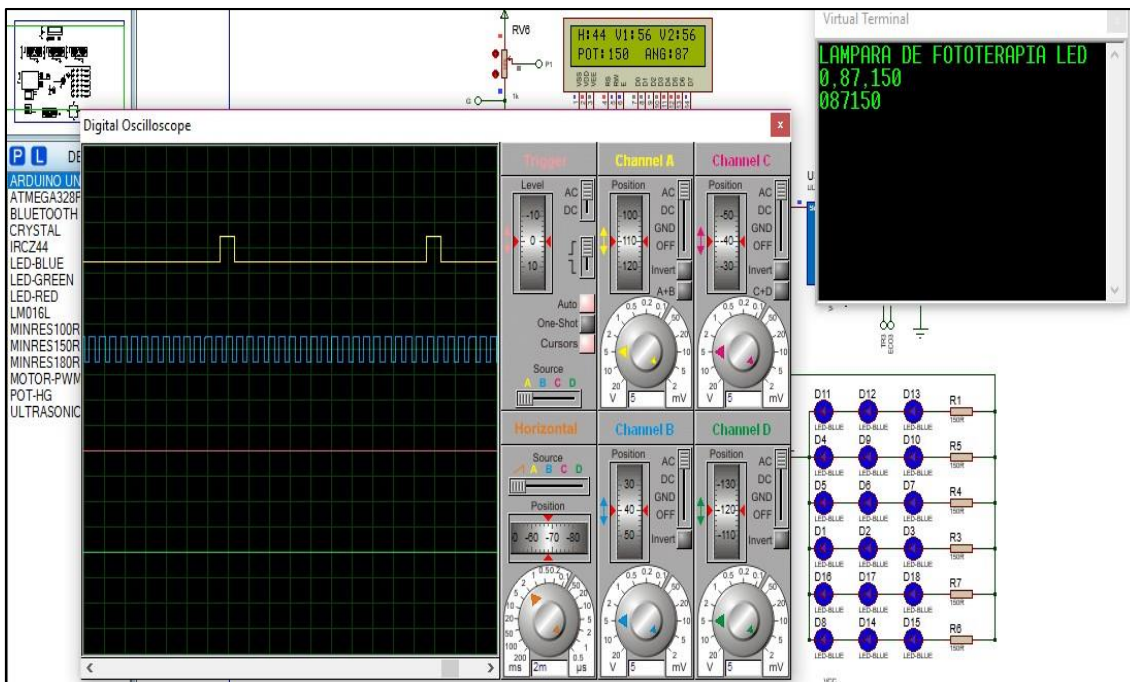


Figura 89: Visualizando las señales PWM para Potencia (150) y ángulo (87°) del servomotor controlado por puerto serial USB. [PROPIO]

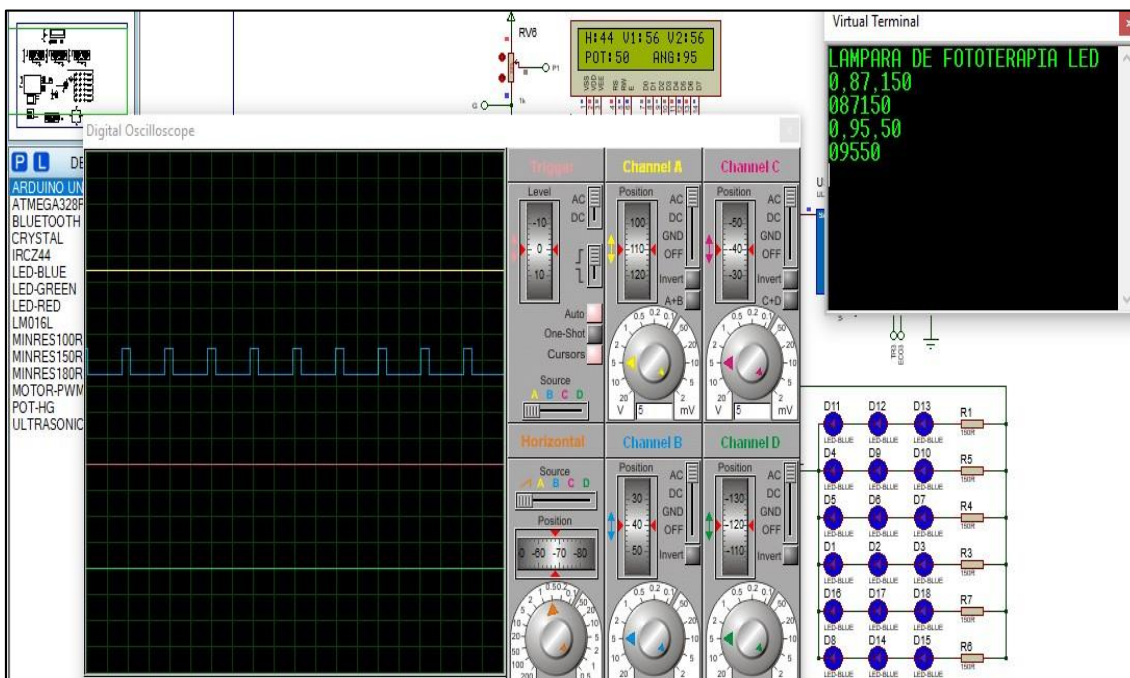


Figura 90: Visualizando las señales PWM para Potencia (50) y ángulo (95°) del servomotor controlado por puerto serial USB. [PROPIO]

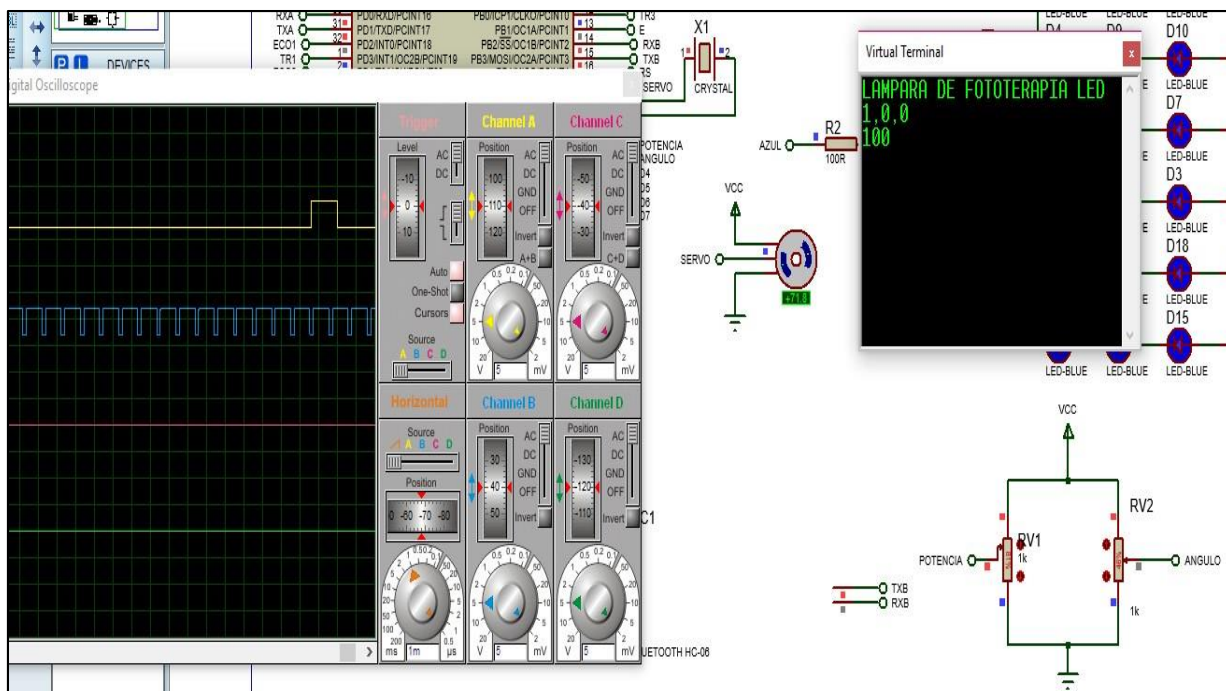


Figura 91: Visualizando las señales PWM para Potencia y ángulo del servomotor controlado por potenciómetro [PROPIO]

4.6. COSTOS FINALES DEL PROYECTO

Todo el proyecto se ha financiado con recursos propios, es así que a continuación presentamos cuadros resumen.

Item	Descripción	Precio Unit.	Cantidad	Total
1	ARDUINO UNO RV3	35	1	35
2	TIRA LED RGB SMD 5050	120	1	120
3	SERVO MOTOR 4KG	30	1	30
4	MODULO BLUETOOTH HC06	20	1	20
5	MOS FET POTENCIA	10	1	10
7	POTENCIOMETROS LINEALES	10	2	20
8	PANTALLA LCD 16X2	25	1	25
9	SENSOR ULTRASONIDO	20	3	60
10	TARJETA PCB	100	1	100
11	CAJA PARA TARJETA ELECTRONICA	30	1	30
12	ESTRUCTURA LAMPARA	130	1	130
13	IMPREVISTOS	100	1	100
				S/ 680.00

Tabla 13: Tabla de costos finales para realización de un prototipo

CAPÍTULO 5: CONCLUSIONES

- Se logró diseñar un prototipo de lámpara de fototerapia led con control de luminosidad para el tratamiento de ictericia en neonatos.
- Se logró estudiar y entender el funcionamiento del sistema de Fototerapia para el tratamiento de la ICTERICIA en neonatos.
- Se logra seleccionar el microcontrolador ATMEGA328 ARDUINO UNO para la implementación del sistema de Fototerapia.
- Se logra diseñar el controlador de corriente para el manejo de la potencia o luminosidad de los leds, así como el programa para el microcontrolador.
- Se logra seleccionar componentes para sistema Bluetooth usando el HC06.
- Se logra desarrollar el software para poder comunicar mediante puerto USB al usuario con una computadora personal.
- Se logra desarrollar el software para poder comunicar mediante bluetooth al usuario con un dispositivo móvil.
- Se logra realizar pruebas y/o simulaciones al sistema de lámpara de fototerapia LED.

CAPÍTULO 6: BIBLIOGRAFÍA

- [1] Tratamiento de la Hiperbilirubinemia (Ictericia) Neonatal, Cristina Burelo Contreras, Universidad Autónoma de Queretaro, Mexico, 10 de Junio del 2011.
<http://infotallerdiseno.files.wordpress.com/2011/12/bili.pdf>
- [2] La irradiación de los aparatos de fototerapia en las Maternidades de maceió, Anne Laura Costa Ferreira, Renata Medeiros do Nascimento, Regina Célia Sales Santos Veríssimo, Universida de Federal de Alagoas, Brasil, Septiembre del 2009.
- [3] Protocolos diagnóstico-terapéuticos de Urgencias Pediátricas SEUP-AEP, Ictericia, Isabel Pinto Fuentes, Servicio de Pediatría Hospital Severo Ochoa, Leganés.
<http://www.aeped.es/sites/default/files/documentos/ictericia.pdf>
- [4] Comparación de dos métodos diagnósticos de ictericia neonatal, MSc. Dra. Ana Campo González, MSc. Dra. Rosa María Alonso Uría, MSc. Dr. Rafael Amador Morán, MSc. Dra. Irka Ballesté López, Facultad de Ciencias Médicas "Dr. Miguel Enríquez". La Habana, Cuba. 2012.
<http://new.medigraphic.com/cgi-bin/resumen.cgi?IDREVISTA=263&IDARTICULO=34680&IDPUBLICACION=3>
- [5] Ictericia Neoatal, Ecuared, Santiago Valdés Martín, Anabel Gómez Vasallo. Temas de Pediatría. Edt. Ciencias Médicas. La Habana, 2006; 393(RR-11):131-132.
- [6] Hiperbilirrubinemia en el recién nacido; Ogas, Marcela del Valle; Campos, Andrea Campos y Ramacciotti, Susana; 2006.
http://www.clinicapediatria.fcm.unc.edu.ar/biblioteca/revisiones_monografias/revisiones/Hiperbilirrubinemia%20en%20el%20Recien%20Nacido.pdf
- [7] Hiperbilirrubinemia neonatal, p.Q.F.B. Lorena Carina García Baeza, Universidad Michoacana De San Nicolás De Hidalgo, Morelia, Michoacán, 2010.
- [8] Guías de practica clinica basadas en la evidencia ictericia neonatal; Dr. Rodrigo Vásquez de Kartzow, Dra. María Ximena Martínez Orozco, Dr. Francisco Acosta Argoti, Dr. Diego Velasco López; Asociacion Colombiana de Facultades de Medicina- ASCOFAME
<http://www.consultorsalud.com/biblioteca/Guias/Ictericia%20Neonatal.pdf>
- [9] Ictericia Neonatal: Revisión; Juliana Carmen Parodi, José Lucio Meana Ibarra, José Horacio Ramos Cosimi Dra. Olga Arce. Médica Neonatóloga – Servicio Neonatología Hospital “JR Vidal”.

- [10] Metabolismo de la bilirrubina, Mónica Angélica Herrera Urbiola, archivo ppt.
<http://s3.amazonaws.com/ppt-download/bilirrubina-110619144955phpapp02.pptx?response-contentdisposition=attachment&Signature=w5QHcRU1Td6B%2BHmdOBYrCDjj3Yw%3D&Expires=1357925792&AWSAccessKeyId=AKIAIW74DRRRQS04NIKA>
- [11] Bioquímica hepática, Metabolismo de los ácidos biliares y la bilirrubina, Prof. J.V. Castell, http://www.uv.es/jcastell/6_Acidos_biliares_y_bilirrubina.pdf
- [12] Síndrome Ictérico Neonatal, Lic.Mat. Rodrigo Neira Contreras, Escuela de Obstetricia y Puericultura Universidad de Chile.
- [13] Artículo Metabolismo de la Bilirrubina, escrito por Meddics, Universidad Nacional Mayor de San Marcos, 2003. <http://www.meddics.com/apuntes-yresumenes/medicina/metabolismo-de-la-bilirrubina.html>
- [14] Metabolismo de la Bilirrubina, Farmacología para todos, 09 Agosto 2011, <http://www.farmacologiaparatodos.com/anatomia/item/64-metabolismo-de-labilirrubina>
- [15] Fototerapia para Ictericia Neonatal, Jeffrey Maisels, M.;McDonagh, Antony F., Revista del Hospital Materno Infantil Ramón Sardá, Vol. 27, Núm. 3, sin mes, 2008,pp. 100-111 Asociación de Profesionales del Hospital Materno Infantil Ramón Sardá Argentina,
<http://redalyc.uaemex.mx/src/inicio/ArtPdfRed.jsp?iCve=91227302>
- [16] Fundamenentos de la fototerapia en la ictericia neonatal, Laura A. Stokowski, RN, MS; Mary Short, RN, MSN; Catherine L. Witt, MS, RNC, NNP, Advances in Neonatal Care Febrero 2007,
http://www.prematuros.cl/webenfermerianeonatal/abril07/Fototerapia/fundamentos_fototerapia.htm
- [17] Beneficios de la Luz Solar Un Punto Brillante para la Salud Humana, Ciencia & Trabajo, Marzo 2009, www.cienciaytrabajo.cl
- [18] Niels RybergFinsen, Biblioteca y Museo Historicomédicos, Universidad de Valencia España, <http://www.historiadelamedicina.org/finsen.htm>
- [19] Fototerapia Con Rayos Ultravioleta, Octavio Small Arana, Revista Peruana de

- Dermatología, Vol. 12, 2002,
http://sisbib.unmsm.edu.pe/bvrevistas/dermatologia/v12_n2/fototerapia_ultravioleta.htm
- [20] Respuesta de las dermatosis tratadas Con fototerapia UVB de banda angosta En el Hospital Nacional Edgardo Rebagliati Martins, Dermatología Peruana, 2007, Vol 17,
http://sisbib.unmsm.edu.pe/bvrevistas/dermatologia/v17_n2/pdf/a04v17n2.pdf
- [21] Guía Tecnológica No.3: Unidad de Fototerapia, Secretaría de salud, Subsecretaría de innovación y calidad, Agosto del 2005, Mexico,
http://www.cenetec.salud.gob.mx/descargas/equipo_guias/guias_tec/3gt_fototerapia.pdf
- [22] Hiperbilirrubinemia en el recién nacido a término, Miriam Martínez-Biarge Y Alfredo García-Alix, Servicio de Neonatología. Hospital Universitario La Paz. Madrid. España.
- [23] Protocolo de fototerapia en Neonatos, Rosa María Ossorio Martínez, Nieves Martín González, Cándida Sánchez González, Consuelo Martínez Jarabo, Juana Ribera Reboloso, Complejo Hospitalario Universitario de Albacete, 2011.
- [24] Lighting, Adrian Javier Leon, Atlantis International University, Honolulu, Hawaii, 2007,
<http://www.aiu.edu/applications/DocumentLibraryManager/upload/Lighting.pdf>
- [25] Iluminación, Ingeniería Rural, Escuela Universitaria de Ingeniería Técnica Agrícola de Ciudad Real,
http://www.uclm.es/area/ing_rural/Instalaciones/Luminotecnia.pdf
- [26] El conocimiento de la luz, Curiosidad científica, 3 de marzo del 2010,
<http://curiosidadcientifica.wordpress.com/2010/03/03/estrellas-el-conocimiento-de-la-luz-i/>
- [27] Manual de luminotecnia para interiores, Carlos Laszlo,
http://www.laszlo.com.ar/Items/ManLumi/issue/Manual_de_Luminotecnia.PDF
- [28] Luminotecnia: Magnitudes fotométricas básicas. Unidades de medida, Blanca Giménez, Vicente Castilla Cabanes, Nuria Martínez Antón, Alicia Pastor Villa, Rosa María, E.T.S. Arquitectura departamento de construcciones arquitectónicas,
<http://riunet.upv.es/bitstream/handle/10251/12732/L%20U%20M%20I%20N%20O%20T%20E%20C%20N%20I%20A.pdf?sequence=1>

- [29] Luminotecnia Básica Para Ingenieros. Antonio Espín Estrella.
<http://www.rebecavega.es/wp-content/uploads/2012/06/TEMAS-1-2-Y-3.pdf>
- [30] Wolfe, Johny chap., Brandy, Globos y Luces: Ami Arganda, 1750-1803
Southern Illinois University, (1999),
<http://es.wikipedia.org/wiki/Quinqu%C3%A9>
- [31] Historia de la iluminación, Andrés Antonio Gil Martín, Innovación y
expectativas educativas, Octubre del 2009, http://www.csicsif.es/andalucia/modules/mod_ense/revista/pdf/Numero_23/ANDRES_A_GIL_1.pdf
- [32] Luminotecnia: Dispositivos para alumbrado incandescente y fluorescente,
Unidad 8, <http://www.mcgraw-hill.es/bcv/guide/capitulo/8448171721.pdf>
- [33] Guía luminotecnia, ERCO, Edición: 01.12.2011,
http://www.ereco.com/download/data/30_media/25_guide_pdf/130_es/es_ereco_guide_6_lighting_technology.pdf
- [34] Luminotecnia, capítulo 12, Ing. Julio Álvarez, Diciembre 2009,
http://www.frba.utn.edu.ar/html/Electrica/archivos/Apuntes_EyM/Capitulo_12_Luminotecnia.pdf
- [35] Narva: Especialistas en luz, Empresa en Brand-Erbisdorf, Berlín, Alemania,
http://www.narvabel.de/es/Productos/Lmparas_fluorescentes_LT/T8_1328.html?sid=es
- [36] Luminotecnia 1: Fuentes luminosas, Universidad Nacional de la Patagonia,
[http://www.ing.unp.edu.ar/asignaturas/maqeinstelectricas/pdf%20\(transparencias\)/Luminotecnia%201.pdf](http://www.ing.unp.edu.ar/asignaturas/maqeinstelectricas/pdf%20(transparencias)/Luminotecnia%201.pdf)
- [37] Dialux, Simulaciones hechas con el catalogo de SITECO, Enero, 2013.
- [38] Diseño y construcción de un sistema de alimentación para un arreglo rgb de tres
led de potencia, Julio Alfredo Cortés Rodríguez, Universidad tecnológica de la
Mixteca, Huajuapán de León, Oaxaca, agosto de 2009, pp. 23
- [39] Light Emitting Diodes (LEDs), ELE 432 Assignment # 3, Vijay Kumar Peddinti,
<http://www.ele.uri.edu/courses/e432/spring08/LEDs.pdf>
- [40] Diodo LED, Alexander Aventura, Cesar Alza, Lima, Perú, 2008,
<http://www.monografias.com/trabajos60/diodo-led/diodo-led2.shtml>
- [41] Funcionamiento de un LED, Desarrollo Lumínico y tecnología LED,
Septiembre, 2011, <http://delumled.blogspot.com/2011/09/funcionamiento-de-un-led.html>

- [42] PHILIPS y la tecnología LED, PHILIPS, 2000,
http://www.lighting.philips.es/pwc_li/es_es/connect/events/assets/LEDs_Toledo/dos sier%20led.pdf
- [43] LED Colors Characteristics, Energy efficiency and renewable energy, U. S. Departament of Energy, Enero, 2012,
<http://apps1.eere.energy.gov/buildings/publications/pdfs/ssl/led-color-characteristicsfactsheet.pdf>
- [44] Historia del LED, Led passion, Marzo, 2011, <http://ledpassion.com/blogled/historia-del-led.html>
- [45] Tecnología LED y su uso en luces de señalización, IALA, Edicion 1, Diciembre, 2005, http://www.iala-aism.org/iala/publications/documentspdf/doc_160_sp.pdf
- [46] Influencia e importancia de la temperatura en los LEDs, 1º Jornada Técnica sobre LEDs, D. Francisco Cavaller, Carandini, Madrid, Octubre , 2010,
http://www.carandini.com/uploads/noticias/1_jornada_LEDs_4_Influencia_temperatura_CARANDINI.pdf
- [47] Lighting emitting diodes, E. F. Schubert,
<http://www.ecse.rpi.edu/~schubert/Course-ECSE-6290%20SDM-2/1%20LEDs.pdf>
- [48] Diseño y construcción de un sistema de alimentación para LEDs de potencia utilizando el convertidor CD/CD tipo Flyback, Victor Armando Bolaños Rodriguez, Universidad Tecnológica de la Mixteca, Huajuapán de León, Febrero, 2009.
- [49] SOMA TECHNOLOGY inc. Wordwide Medical Equipment Sales and Service,
<http://www.somatechnology.com/MedicalProducts>
- [50] Pesos y estatura del bebé, niño y niña.
http://www.guiainfantil.com/salud/embarazo/tabla_pesos.htm
- [51] Comparativa de equipos de fototerapia basados en LEDs, Sergio m. Blanco, Facultades de Ingeniería y Medicina Universidad de la República O. del Uruguay, 3 de abril del 2006, pp. 2
- [52] Fundamentos de óptica, Tema 4: Radiometría y fotometría, Prof. Dr. E. Gómez González, Departamento de Física aplicada III, E. S. Ingenieros -Universidad de Sevilla, 2008-2009
- [53] Física Avanzada: Radiometría y fotometría, Universidad de Vigo. Departamento de Física aplicada, E.T.S. de Ingenieros de Telecomunicación.

- [54] Ángulo, José. **Microcontroladores PIC. Diseño práctico de aplicaciones.** Segunda parte: PIC16F87X. España, McGraw-Hill / Interamericana de España, S.A.U., 2000, 232 pp.
- [55] Barra, O., Barra, F. **Microcontroladores PIC con programación PBP.** España, Editorial RA-MA, 2011, 457 pp.
- [56] Caballero, Alejandro. Metodología de la investigación científica: Diseño con hipótesis explicativas. Lima, UDEGRAF, 2000, 296 pp.
- [57] Hernández, Roberto, Fernández, Carlos y Baptista, Pilar. **Metodología de la Investigación.** 3a ed., México, Ed. McGraw-Hill, 2003, 705 pp.
- [58] Velásquez, Ángel y Rey, Nériida. **Metodología de la investigación científica.** Lima, San Marcos, s/f, 311 pp.
- [59] Todo sobre tiras leds.
<http://www.afinidadelectrica.com/articulo.php?IdArticulo=277>
- [60] Antony Garcia Gonzales. ¿Qué es y cómo funciona un Servomotor?.
<http://panamahitek.com/que-es-y-como funciona-un-servomotor/>
- [61] Diego Pérez de Diego, Sensores de distancia por ultrasonido.
http://picmania.garcia-cuervo.net/recursos/redpictutorials/sensores/sensores_de_distancias_con_ultrasonidos.pdf
- [62] Bluetooth. <https://es.wikipedia.org/wiki/Bluetooth>
- [63] Data Sheet HC06.
http://www.fecegypt.com/uploads/dataSheet/1480849570_hc06.pdf

ANEXOS



Servomotor FUTABA

[PROD 013A- Servomotor Futaba -04/15]

Especificaciones Técnicas del Servomotor Futaba

Diseñado por:

N-MADA®

Características

- Engranajes de Nylon
- Voltaje de operación 4.8-6 Volts.
- Peso 38 gr.
- Velocidad de trabajo
 - o 0.23 seg/60 grados (4.8Volts)
 - o 0.19 seg/60 grados (6 Volts)
- Torque
 - o 3.2 kg/cm (4.8 volts)
 - o 4.1 kg/cm (6 Volts)
- Frecuencia de funcionamiento 50Hz
- Incluye accesorios

Dimensiones (en milímetros)

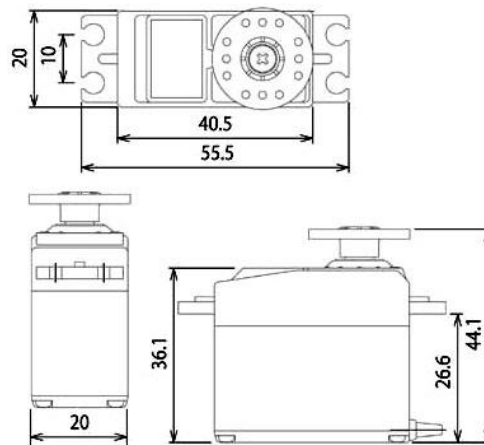


Figura 1. Dimensiones del Servomotor FUTABA [S3003].

Descripción de los pines

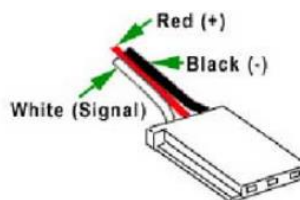


Figura 1. Conector del Servomotor FUTABA [S3003].

Pin	Función
Negro	Tierra
Rojo	Alimentación
Blanco	Control (PWM)

Tabla 1. Descripción del conector del Servomotor FUTABA [S3003].

Código de muestra con NoMADA®

El código de muestra y librerías para este módulo y muchos más se encuentran disponibles en el siguiente enlace en la sección de recursos NoMADA®:

<http://nomada-e.com/videotutoriales.html>

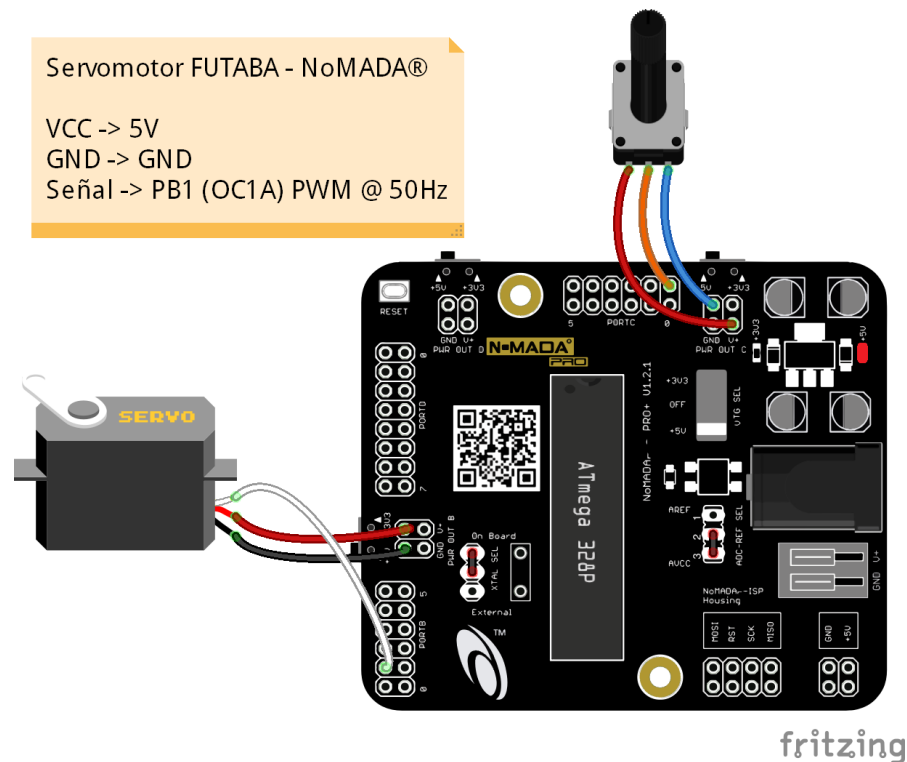


Figura 2. Conexión del Servomotor FUTABA [S3003] con NoMADA®.

Historial de Revisiones del Archivo.

PROD 013A Servomotor FUTABA [S3003] 04/15

1. Revisión Inicial



Diseño Embebido...

un Paso Adelante

*Av. Francisco García Salinas #329 Int. 9
Col. Lomas del Convento
01 492 491 36 82
Guadalupe, Zacatecas, México.*

©2015 NoMADA Store. Todos los derechos reservados | PROD 013A –Servomotor FUTABA [S3003] - 04/2015.



Practica 1:Comenzando con Arduino

1.1 Objetivos.

- Introducción al Arduino.
- Primeros pasos con Arduino.
- Funciones básicas para su programación.
- Ejemplo de aplicación.

1.2 Materiales empleados.

- Arduino UNO.
- Diodo Led.

1.3 Introducción al Arduino.

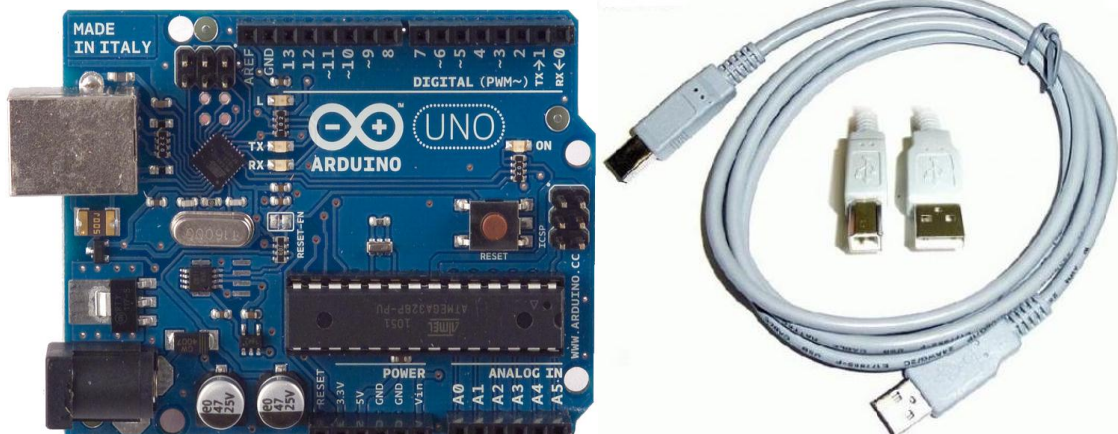
¿Qué es Arduino?

Arduino es una herramienta para hacer que los ordenadores puedan sentir y controlar el mundo físico a través de tu ordenador personal. Es una plataforma de desarrollo de computación física (physical computing) de código abierto, basada en una placa con un sencillo microcontrolador y un entorno de desarrollo para crear software (programas) para la placa.

Puedes usar Arduino para crear objetos interactivos, leyendo datos de una gran variedad de interruptores y sensores y controlar multitud de tipos de luces, motores y otros actuadores físicos. Los proyectos con Arduino pueden ser autónomos o comunicarse con un programa (software) que se ejecute en tu ordenador. La placa puedes montarla tú mismo o comprarla ya lista para usar, y el software de desarrollo es abierto y lo puedes descargar gratis desde la página www.arduino.cc/en/.

El Arduino puede ser alimentado a través de la conexión USB o con una fuente de alimentación externa. La fuente de alimentación se selecciona automáticamente.

Hardware y cable USB

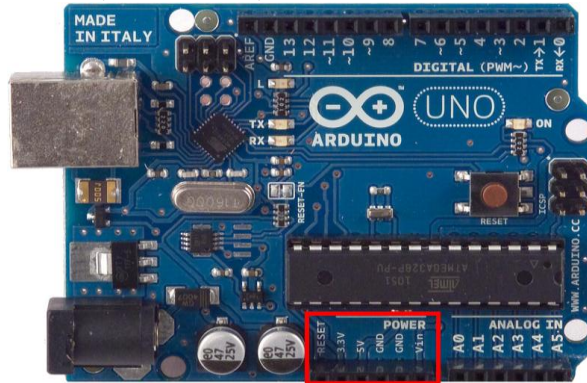


Especificaciones técnicas

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current for I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328)
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz

Power, Inputs and Outputs.

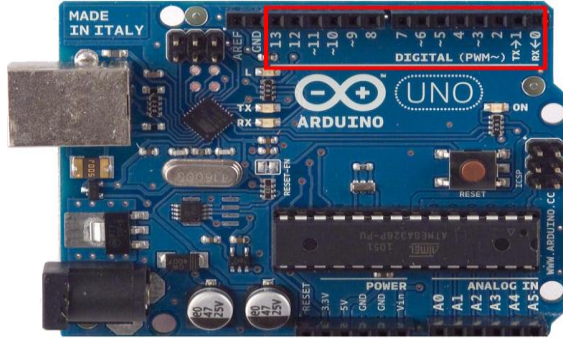
Pines de alimentation (Power Pins)



Bien alimentemos al arduino mediante la conexión USB o mediante una fuente externa (recomendada de 7-12V), vamos a tener unas salidas de tensión continua debido a unos reguladores de tensión y condensadores de estabilización. Estos pines son:

- VIN: se trata de la fuente tensión de entrada que contendrá la tensión a la que estamos alimentando al Arduino mediante la fuente externa.
- 5V: fuente de tensión regulada de 5V, esta tensión puede venir ya sea de pin VIN a través de un regulador interno, o se suministra a través de USB o de otra fuente de 5V regulada.
- 3.3V: fuente de 3.3 voltios generados por el regulador interno con un consumo máximo de corriente de 50mA.
- GND: pines de tierra.

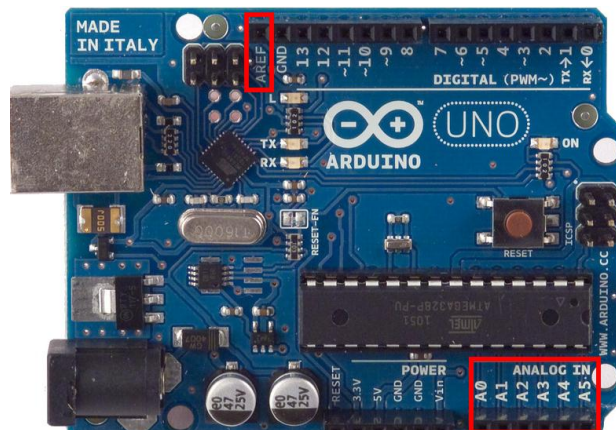
Digital Inputs/Outputs



Cada uno de los 14 pines digitales se puede utilizar como una entrada o salida. Cada pin puede proporcionar o recibir un máximo de 40 mA y tiene una resistencia de pull-up (desconectado por defecto) de 20 a 50 kOhm. Además, algunos pines tienen funciones especializadas como:

- Pin 0 (RX) y 1 (TX). Se utiliza para recibir (RX) y la transmisión (TX) de datos serie TTL.
- Pin 2 y 3. Interrupciones externas. Se trata de pines encargados de interrumpir el programa secuencial establecido por el usuario.
- Pin 3, 5, 6, 9, 10 y 11. PWM (modulación por ancho de pulso). Constituyen 8 bits de salida PWM con la función `analogWrite()`.
- Pin 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Estos pines son de apoyo a la comunicación SPI.
- Pin 13. LED. Hay un LED conectado al pin digital 13. Cuando el pin es de alto valor, el LED está encendido, cuando el valor está bajo, es apagado.

Analog Inputs



El Arduino posee 6 entradas analógicas, etiquetadas desde la A0 a A5, cada una de las cuales ofrecen 10 bits de resolución (es decir, 1024 estados). Por defecto, tenemos una tensión de 5V, pero podemos cambiar este rango utilizando el pin de AREF y utilizando la función `analogReference()`, donde le introducimos una señal externa de continua que la utilizara como referencia.

1.4 Primeros pasos con Arduino

Descarga del IDE (Software) de Arduino

Una vez que conocemos todo los pines necesarios para nuestro manejo y control del Arduino, vamos a instalar el software para poder programarlo mediante el ordenador.

Contamos la placa Arduino

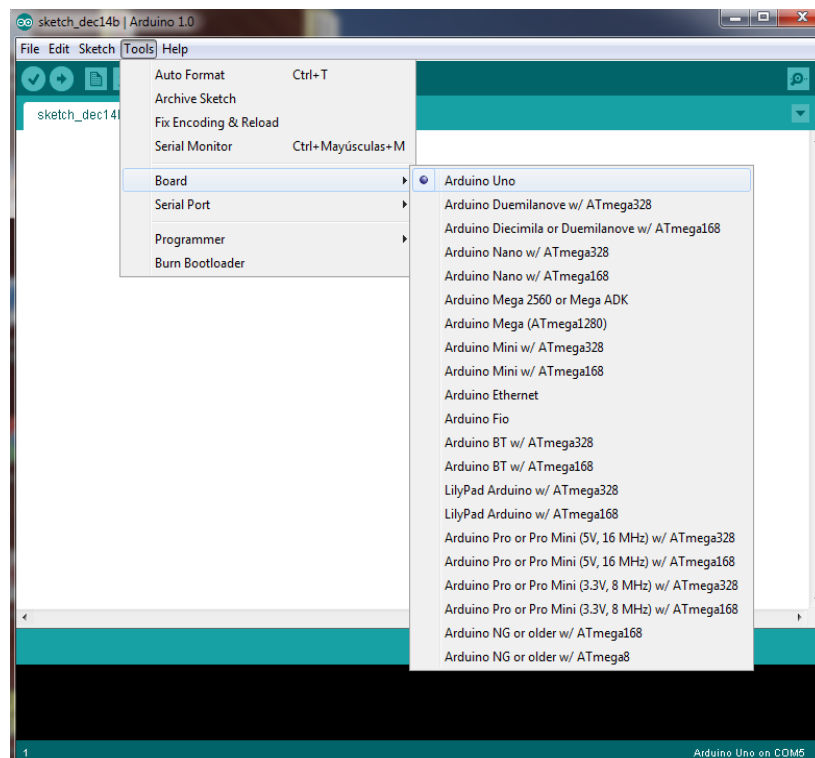
Conectamos la placa Arduino al ordenador usando el cable USB, una vez conectada el led de la placa PWR (led de alimentación) deberá permanecer encendido a partir de ahora.

Instalamos los drivers

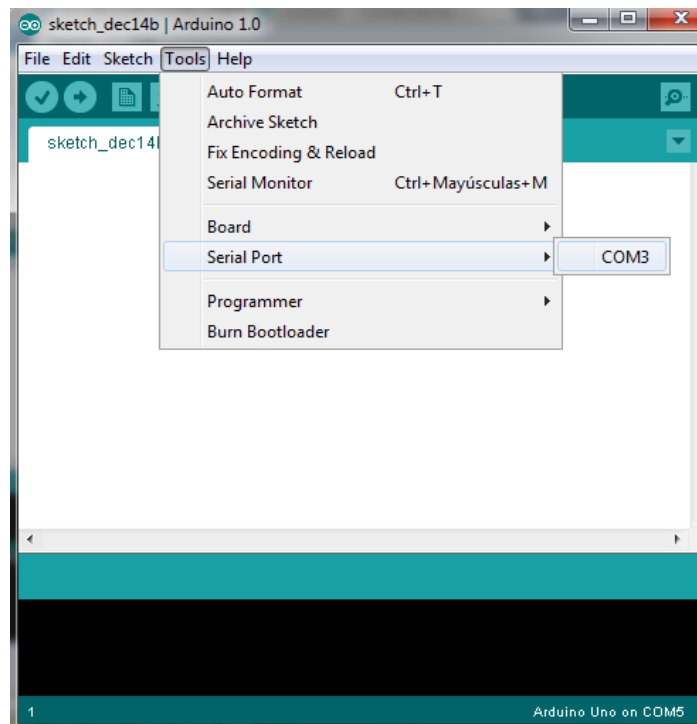
Al conectar el Arduino, Windows automáticamente deberá de inicializar la instalación de los drivers.

Ejecutamos la aplicación Arduino, seleccionamos la placa y el puerto serie.

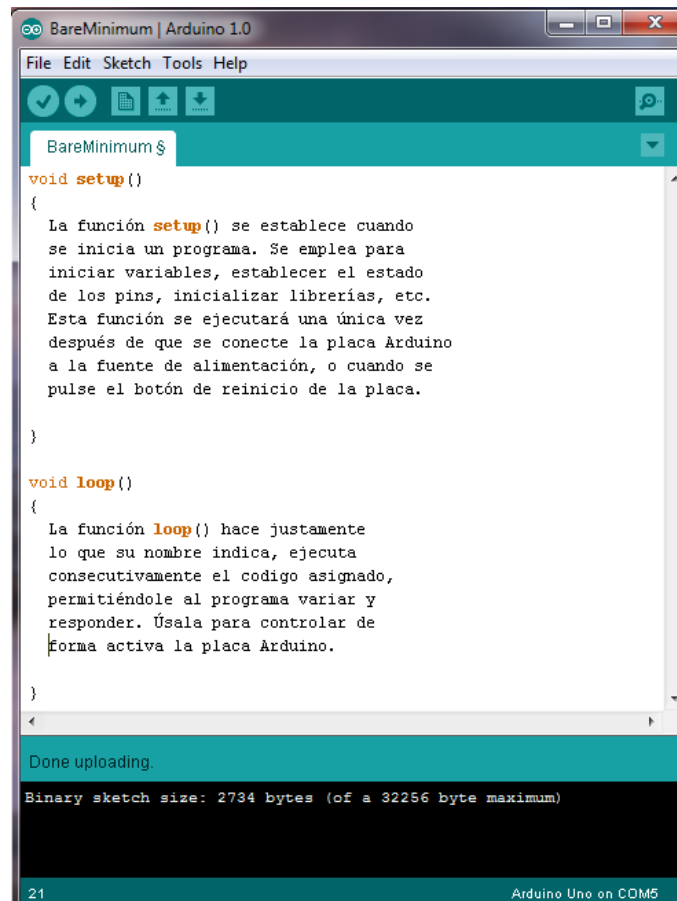
Una vez abierta la aplicación nos vamos a Tools→Board→Arduino UNO



Una vez seleccionado el modelo de nuestra placa tendremos que seleccionar el dispositivo serie de la placa:

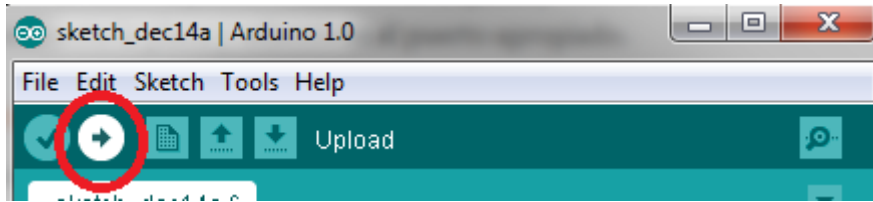


Una vez que tenemos configurada nuestra placa Arduino al ordenador, vamos a estudiar la estructura del lenguaje de programación de Arduino.



Cargar el programa a la placa.

Una vez que tenemos desarrollado el programa completo para cargarlo en el Arduino solo tenemos que:



1.5 Funciones básicas.

- **E/S Digitales**

pinMode(pin,modo)

Configura el pin especificado para comportarse como una entrada (INPUT) o una salida (OUTPUT).

Ejm: `pinMode(Pin13, OUTPUT)`

digitalWrite(pin,valor)

Asigna el valor HIGH (5V) o LOW (0V) a un pin digital.

Ejm: `digitalWrite(Pin13 , HIGH);`

digitalRead(pin)

Lee el valor de un pin digital especificado, HIGH o LOW.

Ejm: `val = digitalRead(Pin13);`

- **E/S Analógicas**

analogRead(pin)

Lee el valor de tensión en el pin analógico especificado. La placa Arduino posee 6 canales conectados a un conversor analógico digital de 10 bits. Esto significa que convertirá tensiones entre 0 y 5 voltios a un número entero entre 0 y 1023. Esto proporciona una resolución en la lectura de: 5 voltios / 1024 unidades, es decir, 0.0049 voltios (4.9mV) por unidad. El rango de entrada puede ser cambiado usando la función [analogReference\(\)](#).

Ejm: `val = analogRead(Pin3)`

analogWrite(pin,valor)

Escribe un valor analógico (PWM) en un pin. Puede ser usado para controlar la luminosidad de un LED o la velocidad de un motor. Después de llamar a la función **analogWrite()**, el pin generará una onda cuadrada estable con el ciclo de trabajo especificado hasta que se vuelva a llamar a la función **analogWrite()** (o una llamada a las funciones **digitalRead()** o **digitalWrite()** en el mismo pin). La frecuencia de la señal PWM será de aproximadamente 490 Hz. los valores de analogRead van desde 0 a 1023 y los valores de analogWrite van desde 0 a 255

Parametros:

- ♦ pin: Es el pin en el cual se quiere generar la señal PWM.
- ♦ valor: El ciclo de trabajo deseado comprendido entre 0 (siempre apagado) y 255 (siempre encendido).

```
Ejm:    val = analogRead(analogPin);  
        analogWrite(ledPin, val / 4);
```

- **Comunicación Serie**

Se utiliza para la comunicación entre la placa Arduino y un ordenador u otros dispositivos. Todas las placas Arduino tienen al menos un puerto serie **Serial**. Se comunica a través de los pines digitales 0 (RX) y 1 (TX), así como con el ordenador mediante USB. Por lo tanto, si utilizas estas funciones, no puedes usar los pines 0 y 1 como entrada o salida digital. Puedes utilizar el monitor del puerto serie incorporado en el entorno Arduino para comunicarte con la placa Arduino. Haz clic en el botón del monitor de puerto serie en la barra de herramientas y selecciona la misma velocidad en baudios utilizada en la llamada a begin().

Serial.begin(speed)

Establece la velocidad de datos en bits por segundo (baudios) para la transmisión de datos en serie. Para comunicarse con el ordenador, utilice una de estas velocidades: 300, 1200, 2400, 4800, 9600, 14400, 19200, 28800, 38400, 57600 o 115200.

```
Ejm: Serial.begin(9600);
```

Serial.read()

Lee los datos entrantes del puerto serie.

```
Ejm; Byte = Serial.read();
```

Serial.print(val,[format])

Imprime los datos al puerto serie como texto ASCII.

val: el valor a imprimir - de cualquier tipo

format: especifica la base (formato) a usar; los valores permitidos son BYTE, BIN (binarios o base 2), OCT (octales o base 8), DEC (decimales o base 10), HEX (hexadecimales o base 16). Para números de coma flotante, este parámetro especifica el número de posiciones decimales a usar.

Ejm: Serial.print(78) imprime "78"
Serial.print('N') imprime "N"
Serial.print(78, BYTE) imprime "N"
Serial.print(78, DEC) imprime "78"
Serial.println(1.23456, 0) imprime "1.23"
Serial.println(1.23456, 2) imprime "1.23"

Serial.println(val,[format])

Imprime los datos al puerto serie como texto ASCII seguido de un retorno de carro (ASCII 13, o '\r') y un carácter de avance de línea (ASCII 10, o '\n').

```
Ejm:Serial.println(analogValue);      // imprime como ASCII
decimal
      Serial.println(analogValue, HEX); // imprime como ASCII
hexadecimal
```

Serial.available()

Devuelve el número de bytes (caracteres) disponibles para ser leídos por el puerto serie. Se refiere a datos ya recibidos y disponibles en el buffer de recepción del puerto (que tiene una capacidad de 128 bytes).

```
if (Serial.available() > 0)
{
    //realiza la lectura del puerto serie
}
```

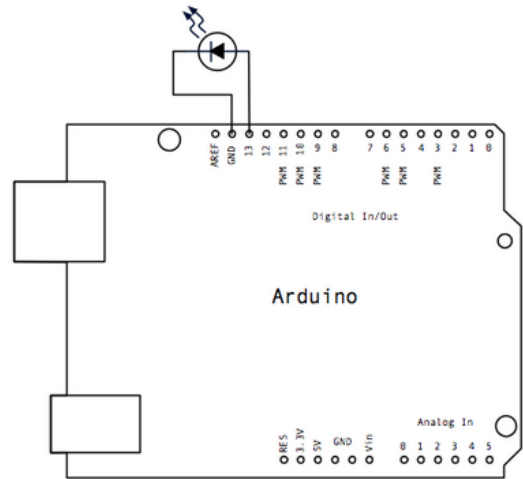
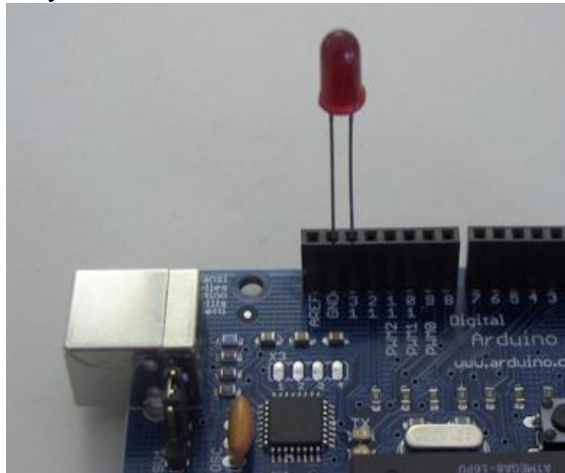
1.6 Ejemplo de Aplicación

Crear un programa que controle el parpadeo de un led, con las funciones anteriormente explicadas, y que se pueda variar el tiempo de estado alto y el tiempo de estado bajo del mismo.

Utilizar la instrucción delay(ms), que hace que el programa se pare y espere el tiempo seleccionado en milisegundos.

Solución:

Podemos asignar la función que encienda y apague a cualquier pin digital, exceptuando el 0 y 1, como comentamos anteriormente, por lo tanto si asignamos el pin 13 como el responsable de esta función, el circuito a montar es muy sencillo:



Y el código de programación para el control del parpadeo del LED sería:

```
/*
PRACTICA 1:
Crear un programa que controle el parpadeo de un led, con
las funciones anteriormente explicadas, y que se pueda
variar el tiempo de estado alto y el tiempo de estado
bajo del mismo.

Utilizar la instrucción delay(ms), que hace que el programa
se pare y espere el tiempo seleccionado en milisegundos.
*/

void setup() {
  // Inicializamos el Pin 13 como una salida, donde va a
  //estar conectado el LED
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH); // Ponemos el LED en estado alto
  delay(1000);             // Tiempo de estado alto en mseg
  digitalWrite(13, LOW);  // Ponemos el LED en estado bajo
  delay(1000);             // Tiempo de estado bajo en mseg
}
```

Practica 6: Coche Digital

6.1 Objetivos

- Conocer aplicaciones más avanzadas sobre Arduino.
- Poner en práctica conocimientos obtenidos en prácticas anteriores.

6.2 Material empleado

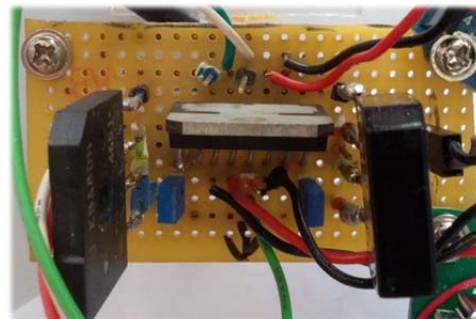
- Arduino UNO.
- Motores eléctricos 1,6W con reductora.
- Controlador dual de puente completo L298N.
- Pantalla LCD.
- Juego de Fotodiodos ópticos de rango visible.

6.3 Componentes del Coche Digital

En este apartado vamos a describir cada uno de los elementos que constituye nuestro coche. Empezamos describiendo los dos motores eléctricos de 1,6W con reductora como los de la siguiente figura:



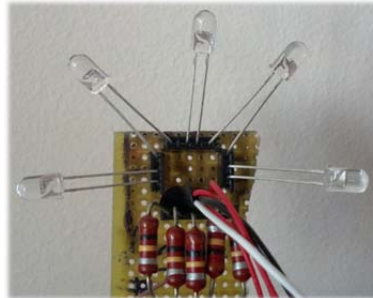
Estos serán los encargados de proporcionar la potencia necesaria a las ruedas para poder realizar el desplazamiento del coche. Estos motores al demandar una potencia que el Arduino es incapaz de dar debemos de conectar entre ambos una etapa de potencia constituida por un Controlador dual de puente completo (L298N) y dos puentes de diodos. La etapa de potencia quedaría como mostramos en la siguiente figura:



A continuación tenemos conectado el Arduino UNO al cual conectaremos la pantalla LCD que nos indicará la dirección que toma el coche en cada momento.

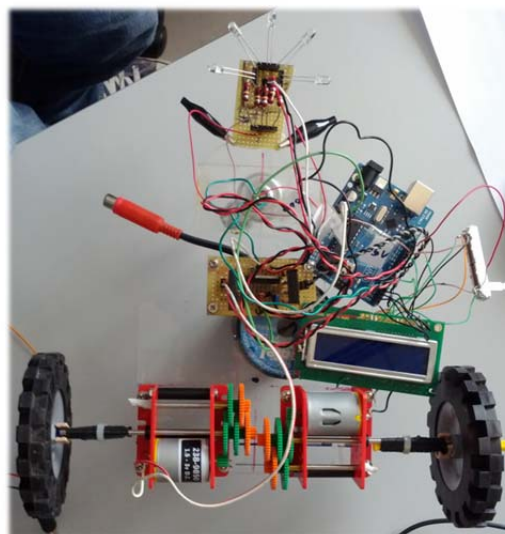


Y por último el juego de fotodiodos que constituirán el “ojo” del coche ya que son los encargados de captar la luz y girar a esta.



Una pequeña descripción de cómo funciona este juego de fotodiodos es que al incidir la luz más a uno que otro, evaluaremos el máximo y ese estará conectado a una entrada digital del Arduino que dará un nivel alto y se desplazará hacia donde ha captado esa luz.

A continuación mostramos una imagen de cómo ha quedado compuesto el coche con todo lo mencionado anteriormente:



6.4 Como crear una librería propia para Arduino.

Para comenzar esta práctica necesitamos crear una librería propia sobre las funciones que utilizaremos para los movimientos del coche. Para ello a continuación iremos explicando paso a paso cómo realizar esta librería.

Para crear una librería se necesita al menos dos archivos: un archivo de cabecera (w / con extensión. H) y el código fuente (w / extensión. cpp). El archivo de cabecera contiene definiciones para la librería: básicamente un listado de todo lo que hay dentro, mientras que el archivo del código fuente tiene el código real. Vamos a llamar a nuestra librería "CocheDigital", por lo que nuestro archivo de cabecera será CocheDigital.h. Echemos un vistazo a lo que contiene. Puede parecer un poco extraño al principio, pero tendrá más sentido una vez que se vea el código fuente que lo acompaña.

El archivo de cabecera consiste básicamente en una clase con una línea para cada función de la librería, junto con las variables que se van a usar:

```
class CocheDigital {
public:
    CocheDigital(int izq_en,int izq_avan,int izq_retro,int der_en,int der_avan,int der_retro);
    void avanza();
    void retrocede();
    void avanza(int tiempo);
    void retrocede(int tiempo);
    void para();
    void frena();
    void derecha();
    void derecha(int tiempo);
    void izquierda();
    void izquierda(int tiempo);
    void derecha_avanza();
    void derecha_avanza(int tiempo);
    void izquierda_avanza();
    void izquierda_avanza(int tiempo);
    void derecha_retrocede();
    void derecha_retrocede(int tiempo);
    void izquierda_retrocede();
    void izquierda_retrocede(int tiempo);
private:
    int _izq_avan;
    int _izq_retro;
    int _der_avan;
    int _der_retro;
    int _izq_en;
    int _der_en;
};
```

Una clase es simplemente una colección de funciones y variables agrupadas en un mismo lugar. Estas funciones y variables pueden ser públicas, lo que significa que las podrán usar las personas que están utilizando la librería, o privada, lo que significa que sólo se puede acceder a ellas desde la propia clase. Cada clase tiene una función especial conocida como constructor, que se utiliza para crear una instancia de la clase (o sea, un objeto). El constructor tiene el mismo nombre que la clase, y no devuelve nada.

Se necesitan un par cosas más en el archivo de encabezado. Una de ellas es una instrucción # include que da acceso a los tipos estándar y las constantes del lenguaje Arduino (esto se agrega automáticamente a los programas normales, pero no a las librerías). Se parece a esto (y se coloca antes de la definición de la clase mostrada anteriormente):

```
#include "Arduino.h"
```

Y por último para tener la cabecera completa necesitamos las dos siguientes líneas de comando, cuya función es preguntar si esta definida la librería y si no lo está la define:

```
#ifndef CocheDigital_h
#define CocheDigital_h

...

#endif
```

Con todo esto ya podemos completar nuestro archivo de cabecera de la librería como mostramos a continuación:

```
/*
CocheDigital.h
Libreria para el control del Coche desarrollado por el
PAIDI-TIC_168 con control digital de los motores.
Abril 2012.
*/

#ifndef CocheDigital_h
#define CocheDigital_h

#include "Arduino.h"

class CocheDigital {

public:
    CocheDigital(int izq_en,int izq_avan,int izq_retro,int der_en,int der_avan,int der_retro);
    void avanza();
    void retrocede();
    void avanza(int tiempo);
    void retrocede(int tiempo);
    void para();
    void frena();
    void derecha();
    void derecha(int tiempo);
    void izquierda();
    void izquierda(int tiempo);
    void derecha_avanza();
    void derecha_avanza(int tiempo);
    void izquierda_avanza();
    void izquierda_avanza(int tiempo);
    void derecha_retrocede();
    void derecha_retrocede(int tiempo);
    void izquierda_retrocede();
    void izquierda_retrocede(int tiempo);
private:
    int _izq_avan;
    int _izq_retro;
    int _der_avan;
    int _der_retro;
    int _izq_en;
    int _der_en;

};

#endif
```


A continuación lo que nos quedaría es explicar las partes que componen el código fuente de CocheDigital.cpp.

Lo primero son un par de # include. Con esto el resto del código tendrá acceso a las funciones estándar de Arduino, y a las definiciones definidas en la cabecera CocheDigital.h.:

```
#include "Arduino.h"
#include "CocheDigital.h"
```

A continuación viene el constructor. Una vez más, en el constructor se establece lo que debe ocurrir cuando alguien crea una instancia de la clase. En este caso, se especifica los pines que vamos a utilizar. Configuramos los pines como salida de las variables privadas para su uso en las otras funciones:

```
CocheDigital::CocheDigital(int izq_en,int izq_avan,int izq_retro,int der_en,int der_avan,int der_retro)
{
    _izq_avan=izq_avan;
    _izq_retro=izq_retro;
    _der_avan=der_avan;
    _der_retro=der_retro;
    _izq_en=izq_en;
    _der_en=der_en;
}
```

Se puede observar un par de cosas extrañas en este código. La primera es el *CocheDigital::* antes del nombre de la función. Esto indica que la función es parte de la clase CocheDigital. Verás esto en otras funciones de la clase. Lo segundo es el subrayado en el nombre de las variables privadas, *_variable*. Esta variable puede tener cualquier nombre, siempre y cuando coincida con la definición que figura en el archivo de encabezado. Se le agrega un subrayado al inicio del nombre dejar claro que las variables son privadas, y también para diferenciarlas del argumento de la función, esto es como norma estándar.

Una vez descrito la primera parte del código fuente vamos al cuerpo donde quedar definidas todas las funciones de nuestra propia librería.

Vamos a indicar dos tipos de funciones que podría realizar el coche y como las hemos introducido en la librería y el resto de ellas sería similares.

Por un lado tenemos la función *avanza()* que es la encargada de que el coche ande en dirección recta por lo cual le damos un valor alto a los pines de salidas digitales (*_izq_avan* y *_der_avan*) que son los que están conectados a ambos motores que moverán las ruedas del coche. El código de la función es:

```
void CocheDigital::avanza()
{
    digitalWrite(_izq_avan,HIGH);
    digitalWrite(_der_avan,HIGH);
    digitalWrite(_izq_retro,LOW);
    digitalWrite(_der_retro,LOW);
    digitalWrite(_izq_en,HIGH);
    digitalWrite(_der_en,HIGH);
}
```

En esta función observamos dos pines habilitados de ENABLE la función de este se ve recogida claramente en la siguiente tabla:

	Avanza	Retrocede	Operación
E=1	1	0	Avanza
	0	1	Retrocede
	1	1	X
E=0	x	x	X

Se observa que hasta que el enable no este activado no va a realizar la función ordenada por eso en la función anterior le damos la orden prevista de que avance y hasta que no se activa el enable no se va a ejecutar.

Otra posible función que hemos creado es la de *avanza(int tiempo)* la diferencia de esta con la anteriormente descrita es que a esta le introducimos el tiempo que queremos que realice la operación correspondiente.

```
void CocheDigital::avanza(int tiempo)
{
    digitalWrite(_izq_avan,HIGH);
    digitalWrite(_der_avan,HIGH);
    digitalWrite(_izq_retro,LOW);
    digitalWrite(_der_retro,LOW);
    digitalWrite(_izq_en,HIGH);
    digitalWrite(_der_en,HIGH);
    delay(tiempo);
    digitalWrite(_izq_en,LOW);
    digitalWrite(_der_en,LOW);
}
```

En esta se observa a diferencia con la anterior que le damos la instrucción correspondiente activamos el enable de ambos motores esperamos el tiempo introducido y desactivamos el enable para que deje de realizar la instrucción ordenada.

Una vez que tenemos creado tanto el archivo de cabecera como el código fuente el siguiente paso será crear el directorio CocheDigital dentro del subdirectorios de librerías de Arduino. Copiamos tanto al fichero *CocheDigital.h* como *CocheDigital.cpp* en ese directorio. Ahora ejecutamos el IDE de Arduino. Vamos al menú Sketch > Import Library y deberíamos ver la opción CocheDigital. Una vez seleccionado la librería ya será compilada con los programas que utiliza. Si no aparece la librería, debemos de asegurarnos de que los nombres de los archivos terminan realmente en .cpp y .h.

Por ultimo para terminar de completar la librería Arduino no es capaz de reconocer las funciones que hemos creado en el cuerpo y no las resalta, por lo que debemos de crear un archivo llamado keywords.txt en el directorio de CocheDigital. Este fichero tendrá este aspecto:

*CocheDigital*KEYWORD1 *avanza*KEYWORD2 *retrocede*KEYWORD2

Cada línea tiene el nombre de la palabra clave, seguido de un tabulador (no espacios), seguido por el tipo de palabra clave. Las clases deben ser del tipo KEYWORD1 y se muestran de color naranja y las funciones deben ser del tipo KEYWORD2 y serán de color marrón.

6.5 Control del Coche basándose en la librería creada

Una vez que sabemos cómo está constituido el coche y hemos creado la librería con todas las funciones posibles, vamos a crear unos programas para su control.

Programa 1:

El primer programa consiste en mediante puerto serie introducir una letra y que el coche realice la operación correspondiente durante un tiempo establecido, utilizando las funciones que tenemos que introducir un tiempo. Las letras por ejemplo pueden establecerse de la siguiente forma:

- a-avanza
- p-para
- d-derecha
- i-izquierda
- r-retrocede
- b-derecha avanza
- c-izquierda avanza
- e-derecha retrocede
- f-izquierda retrocede
- 0-frena

El código del programa junto con la explicación de cada línea sería:

```
//Incluimos la libreria correspondiente
#include <CocheDigital.h>

CocheDigital coche(4,5,6,8,9,10);
/*Pin 4 = izq_en
   Pin 5 = izq_avan
   Pin 6 = izq_retro
   Pin 8 = der_en
   Pin 9 = der_avan
   Pin 10= der_retro*/
/*Declaramos una variable tiempo que sera la
encargada de establecer el tiempo que queremos
que realiza la operacion correspondiente.*/
int tiempo=1000;

void setup()
{
    Serial.begin(9600); //Inicializamos el Puerto
    //Serie ya que es por donde introduciremos el
    //caracter para que realice una o otra operacion.
}
char inByte;
void loop()
{
    if (Serial.available() > 0) { //Comprobamos si el puerto esta disponible
        inByte = Serial.read(); //Guardamos el valor introducido en una variable
        //Y ya ejecutamos la operacion correspondiente dependiendo del caracter
        //introducido, mostrandonos por puerto serie la operacion realizada
    }
}
```

```

if (inByte=='a'){
    Serial.println("Avanza");
    coche.avanza(tiempo);
}
else if (inByte=='p'){
    Serial.println("Para");
    coche.para();
}
    else if (inByte=='d'){
        Serial.println("Derecha");
        coche.derecha(tiempo);
    }
    else if (inByte=='i'){
        Serial.println("Izquierda");
        coche.izquierda(tiempo);
    }
    else if (inByte=='r'){
        Serial.println("Retrocede");
        coche.retrocede(tiempo);
    }
    else if (inByte=='b'){
        Serial.println("Derecha avanza");
        coche.derecha_avanza(tiempo);
    }
    else if (inByte=='c'){
        Serial.println("Izquierda avanza");
        coche.izquierda_avanza(tiempo);
    }
    else if (inByte=='e'){
        Serial.println("Derecha retrocede");
        coche.derecha_retrocede(tiempo);
    }
    else if (inByte=='f'){
        Serial.println("Izquierda retrocede");
        coche.izquierda_retrocede(tiempo);
    }
else if (inByte=='0'){
    Serial.println("Frena");
    coche.frena();
}
else Serial.println("Comando de control incorrecto");
}
}

```

Programa 2:

Ahora este segundo programa vamos a complicar un poco más la cosa y vamos hacer que el coche realice la operación correspondiente por sí solo, para ello como comentamos al principio de la práctica tenemos un juego de fotodiodos que serán los encargados de transmitirle una señal al Arduino que controlara el movimiento de coche dependiendo de la cantidad de luz y el mismo se orienta hacia ella hasta encontrarla y una vez que se enfrente avanza en dirección a la luz.

Como complemento a este problema hemos instalado la pantalla LCD que ya controlamos en prácticas anteriores, y en esta pantalla lo que nos va indicar es la dirección que está tomando el coche en cada momento.

A continuación se muestra el código del programa con la explicación en cada una de las líneas de comando:

```
/*Inicializacion de ambas librerias */
#include <LiquidCrystal.h>
#include <CocheDigital.h>

/*Le damos a ambas Librerias los pines que tienen habilitados */
CocheDigital coche(4,5,6,8,9,10);
/*Pin 4 = izq_en
   Pin 5 = izq_avan
   Pin 6 = izq_retro
   Pin 8 = der_en
   Pin 9 = der_avan
   Pin 10= der_retro*/
LiquidCrystal lcd(11, 7, 3, 2, 1, 0);

void setup()
{
  /*Utilizamos los pines 13 y 12 como fuentes de
  5V para alimentar a la pantalla y al juego de fotodiodos */
  pinMode(13,OUTPUT);
  digitalWrite(13,HIGH);
  pinMode(12,OUTPUT);
  digitalWrite(12,HIGH);

  //Inicializacion de la pantalla 16 filas y 2 columnas
  lcd.begin(16, 2);
  //Mensaje inicial en la pantalla
  lcd.print("ARDUINO");
  delay(1000);
  lcd.clear(); //Limpiamos la pantalla
}
/*Definicion de variables utilizadas */
int Sensor[5];
int maximol,maximo2,maximo;
```

```

void loop()
{
    /*Cada lectura de los fotodiodos la
    introducimos en una posicion del vector Sensor */
    Sensor[0]=analogRead(A0);
    Sensor[1]=analogRead(A1);
    Sensor[2]=analogRead(A2);
    Sensor[3]=analogRead(A3);
    Sensor[4]=analogRead(A4);

    /* Buscamos el maximo entre la tension
    proporcionada por los sensores*/
    maximo1=max(Sensor[0],Sensor[1]);
    maximo2=max(Sensor[2],Sensor[3]);
    maximo=max(maximo1,maximo2);
    maximo=max(maximo,Sensor[4]);

    /*Y dependiendo de donde detecta el maximo de
    tension que corresponde con el maximo de haz de
    luz detectado realizara un movimiento u otro
    Y en la pantalla se escribira el desplazamiento
    correspondiente*/

    if (Sensor[0]==maximo||Sensor[1]==maximo)
    {
        coche.derecha(500);
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("Derecha");
        lcd.setCursor(8, 0);
        lcd.print("<--");
    }
    if (Sensor[3]==maximo||Sensor[4]==maximo)
    {
        coche.izquierda(500);
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("Izquierda");
        lcd.setCursor(10, 0);
        lcd.print("-->");
    }
    if (Sensor[2]==maximo)
    {
        if ((Sensor[1]-Sensor[3])>50)
        {
            coche.derecha(200);
            lcd.clear();
            lcd.setCursor(0, 0);
            lcd.print("Derecha");
            lcd.setCursor(8, 0);
            lcd.print("<--");
        }
    }
}

```

```
else if ((Sensor[1]-Sensor[3])<-50)
{
    coche.izquierda(200);
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Izquierda");
    lcd.setCursor(10, 0);
    lcd.print("<--");
}
else
{
    coche.avanza(200);
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Avanza");
    lcd.setCursor(8, 0);
    lcd.print("|");
    lcd.setCursor(8, 1);
    lcd.print("V");
}
}
}
```


Practica 5: Control de una Pantalla LCD mediante Arduino

5.1 Objetivos

- Control y conocimiento de la hoja de característica de una pantalla LCD.
- Conexión de la pantalla LCD con el Arduino.
- Funciones de la librería LiquidCrystal.
- Aplicación práctica del control de la pantalla mediante Arduino.

5.2 Material Empleado.

- Arduino UNO.
- Pantalla LCD.
- Potenciómetro 10K.
- Pila AAA.

5.3 Hoja de características de la pantalla LCD.

Antes de meterle mano a la programación en el Arduino es necesario conocer algunos parámetros límites que posee la pantalla además de la localización de cada una de las patillas que posteriormente necesitaremos para conectar con el Arduino.

En la Tabla5.1 mostramos los límites básicos que posee la pantalla que utilizaremos en nuestra práctica.

Parameter	Symbol	Min	Max	Unit
Supply voltage for logic	VDD	-0.3	7.0	V
Supply voltage for LCD	VDD - VO	-0.3	VDD+0.3	V
Input voltage	VI	-0.3	VDD+0.3	V
Normal operating temperature	TOP	0	50	°C
Normal storage temperature	TST	-10	60	°C
Wide operating / storage temperature (except FSTN)	TOP / TST	-30	80	°C
Wide operating / storage temperature (FSTN)	TOP / TST	-30	70	°C

Tabla5.1- Límites operativos de la pantalla LCD 162B

En la Figura5.1 podemos observar la localización de las patillas que posee la pantalla, para saber cuál es la patilla 1 y cual la 16.

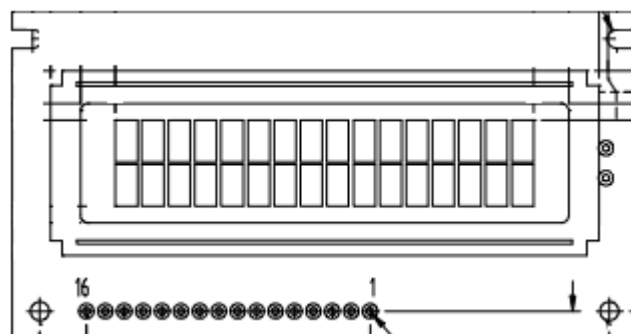


Figura5.1. Localización de las patillas de la pantalla.

Y por último en la Figura5.2 observamos la correspondencia de cada patilla y una explicación posterior de la funcionalidad de cada una de ellas.

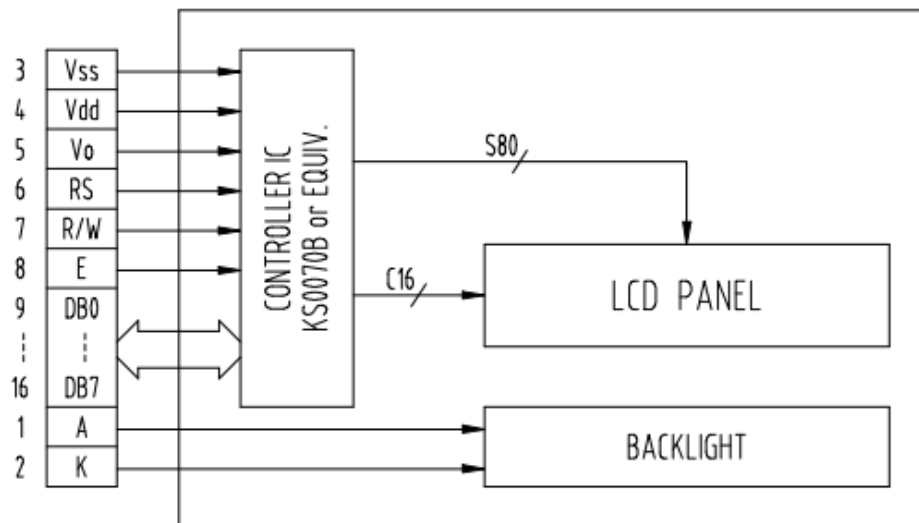


Figura5.2.

El LCD tiene un interfaz paralelo, significando esto que el microcontrolador tiene que manipular varios pines del interfaz a la vez para controlarlo. El interfaz consta de los siguientes pines:

- **Vss:** Tensión GND o de referencia (0V).
- **Vdd:** Tensión de alimentación a la pantalla (+5V).
- **Vo:** Pin de contraste de la pantalla.
- **RS:** Pin de selección de registro que controla en que parte de la memoria del LCD estas escribiendo datos.
- **R/W:** Modo lectura (H) o modo escritura (L).
- **E:** Pin de habilitación de los registros (Enable).
- **DB0...DB7:** se tratan de 8 pines de datos cuyos estados (H o L) son los bits que estas escribiendo a un registro cuando escribes, o los valores de lectura cuando están en el modo lectura.
- **A:** Ánodo del diodo de retro-iluminación (Bklt+).
- **K:** Cátodo del diodo de retro-iluminación (Bklt-).

5.4 Conexionado del LCD con el Arduino

Una vez que tenemos todos los pines del LCD identificados el siguiente paso es el conexionado entre el Arduino y la pantalla como mostramos en la Figura5.3.

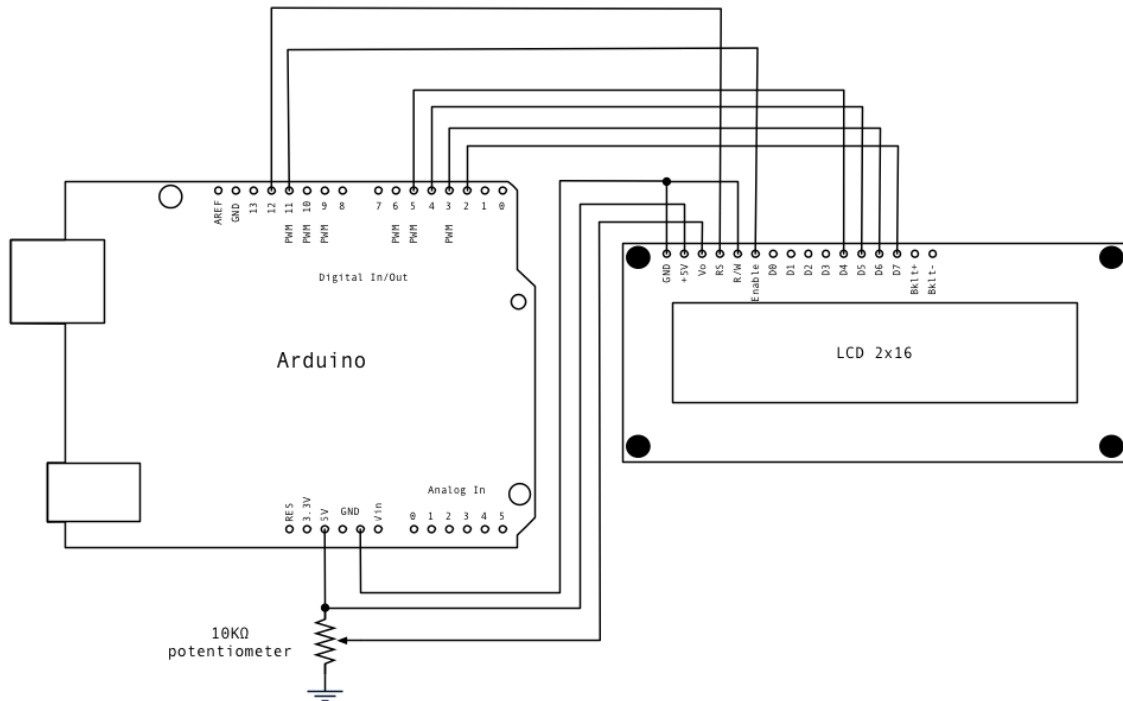


Figura5.3. Conexionado de la placa Arduino con la pantalla LCD.

Se destaca:

- ♦ El pin RS del LCD conectado a la E/S digital en el pin 12
- ♦ El pin enable del LCD conectado a la E/S digital en el pin 11.
- ♦ Los pines D4 - D7 conectado a las E/S digitales desde el pin 5 hasta el 2.
- ♦ Los pines de voltaje y tierra conectados a +5V y tierra.
- ♦ El pin Vo, que controla el contraste, conectado a un potenciómetro. Ajusta el potenciómetro para que el texto tenga el contraste que tú quieras.

5.5 Funciones de la librería LiquidCrystal

Esta biblioteca permite a la placa Arduino controlar displays LCD. La biblioteca trabaja en modo 4-bit o en 8-bit (es decir, por medio de 4 u 8 líneas de datos, además de RS, ENABLE, y, opcionalmente, las líneas de control RW). Entre las funciones básicas a destacar tenemos:

- **LiquidCrystal():**

- Descripción:

Crea una variable de tipo LiquidCrystal. La pantalla se puede controlar por medio de 4 u 8 líneas de datos. En el primer caso, omitir los números de pines para d0 hasta d3 y dejar esos pines no conectados. El pin RW pueden ser conectado a masa en lugar de conectarse a un pin de Arduino, si es así, omítelo de los parámetros de esta función.

- Sintaxis:

LiquidCrystal (rs, rw, enable, d0, d1, d2, d3, d4, d5, d6, d7)

- Parámetros:

- ♦ rs: número del pin de Arduino que está conectado al pin RS del LCD
 - ♦ rw: número del pin de Arduino que está conectado al pin RW del LCD (opcional)
 - ♦ enable: número del pin de Arduino que está conectado al pin ENABLE del LCD
 - ♦ d0, d1, d2, d3, d4, d5, d6, d7: números de pines de Arduino que están conectados a los correspondientes pines de datos del LCD. d0, d1, d2, y d3 son opcionales; si se omiten, el LCD será controlado usando solamente cuatro líneas de datos (d4, d5, d6, d7).

- **Begin()**

- Descripción:

Especifica las dimensiones (ancho y alto) del display LCD.

- Sintaxis:

Lcd.begin(columnas,filas)

- Parámetros:

- ♦ lcd: una variable de tipo LiquidCrystal.
 - ♦ columnas: número de columnas que tiene el display.
 - ♦ filas: número de filas que tiene el display.

- **clear**
 - Descripción:

Borra la pantalla del display LCD y posiciona el cursor en la esquina superior-izquierda.
 - Sintaxis:

`Lcd.clear()`
 - Parámetros:
 - ♦ lcd: una variable de tipo LiquidCrystal.
- **home()**
 - Descripción:

Posiciona el cursor en la esquina superior-izquierda del LCD. Se trata de la posición en la que aparecerá el siguiente texto escrito en el display. A diferencia con al anterior que esta no borra el contenido anterior del display.
 - Sintaxis:

`Lcd.home()`
 - Parámetros:
 - ♦ lcd: una variable de tipo LiquidCrystal.
- **setCursor()**
 - Descripción:

Establece la posición del cursor donde empezaran aparecer los siguientes caracteres escritos en el LCD.
 - Sintaxis:

`Lcd.setCursor(columna, fila)`
 - Parámetros:
 - ♦ lcd: una variable de tipo LiquidCrystal.
 - ♦ columna: columna donde posicionar el cursor (0-primer columna).
 - ♦ fila: fila donde posicionar el cursor (0-primer fila).
- **write()**
 - Descripción:

Escribe un carácter en el LCD.
 - Sintaxis:

`Lcd.write(data)`
 - Parámetros:
 - ♦ lcd: una variable de tipo LiquidCrystal.
 - ♦ data: el carácter a escribir en el display.

- **print()**
 - Descripción:
Imprime un texto en el LCD.
 - Sintaxis:
`Lcd.print(data,BASE)`
 - Parámetros:
 - ♦ lcd: una variable de tipo LiquidCrystal.
 - ♦ data: los datos a imprimir (char, byte, int...).
 - ♦ BASE (opcional): la base en la que se van a imprimir los números: BIN, DEC, OCT, HEX.

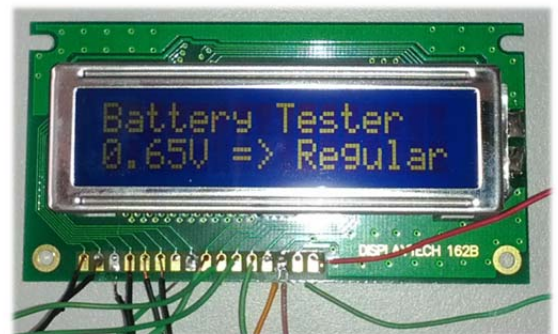
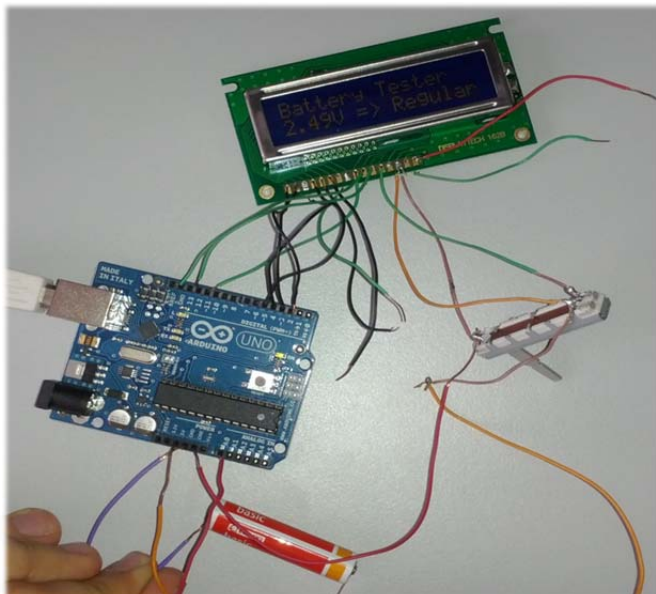
5.6 Ejercicios Prácticos

Ejercicio 1:

Una vez que tenemos conocimiento de cómo conectar nuestra pantalla LCD al Arduino y conocemos las funciones básicas de la librería LiquidCrystal que nos permiten realizar la conexión, como ejercicio se pretende realizar un programa que permita transmitir mediante puerto serie una palabra o texto corto que nosotros escribamos y mostrarlo en el Arduino además de que en una segunda línea de la pantalla nos muestre un contador con el número de caracteres introducidos anteriormente.

Ejercicio 2:

En este segundo ejercicio vamos a modificar el programa anteriormente creado para testear el nivel de tensión de una pila AA o AAA. Se pretenderá conectar una pila entre la entrada analógica 0 y tierra, leer el valor de tensión y mostrar en la pantalla LCD el estado de la pila si es Perfecto ($>1.40V$), Bueno ($[1.20V, 1.40V]$), Regular ($[1.20V, 0.30V]$), Malo ($[<0.30V]$). En la siguiente imagen se observa el resultado de este ejercicio:



Solución Ejercicio1:

```
// Habilitamos la libreria LiquidCrystal
#include <LiquidCrystal.h>

// Inicializacion de los pines a utilizar
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup()
{
    // Inicializamos el numero de filas y columnas de la pantalla
    lcd.begin(16, 2);
    // Inicializamos la comunicacion por puerto serie:
    Serial.begin(9600);
}

void loop()
{
    int cont=0; //Declaracion e inicializacion del contador de palabras
    if (Serial.available())
    {
        delay(100);
        // Limpiamos la pantalla
        lcd.clear();

        // Leemos los caracteres introducidos por puerto serie
        while (Serial.available() > 0)
        {
            // Escribimos en la pantalla LCD
            lcd.write(Serial.read());
            //Contador de caracteres;
            cont=cont+1;
        }
        //Bajamos el cursor a la segunda linea de la pantalla
        lcd.setCursor(0,1);
        //Escribimos el valor del contador en pantalla
        lcd.print(cont);
    }
}
```


Solución Ejercicio 2:

```
/*
  Battery Tester
  Programa que trata de mostrar el voltaje de una pila en una pantalla LCD,
  estando la pila conectada entre la entrada analógica 0 y tierra. La pantalla
  LCD dependiendo del nivel de tensión de la Pila nos mostrará el estado de
  la misma(Perfecto, Bueno, Regular, Malo).
*/

// Habilitamos la librería LiquidCrystal
#include <LiquidCrystal.h>

// Inicialización de los pines a utilizar
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup() {
  // Inicializamos el número de filas y columnas de la pantalla
  lcd.begin(16, 2);
  // Mensaje instantáneo al iniciar el programa
  lcd.print("ARDUINO");
  delay(1000);
  lcd.clear(); //Limpiamos el texto anterior
}

void loop() {
  lcd.clear();
  lcd.setCursor(0, 0); //Ponemos el contador en la posición 0 de la pantalla:
  lcd.print("Battery Tester"); //Escribimos el texto
  lcd.setCursor(0, 1); //bajamos el cursor a la segunda línea
  lcd.print(analogRead(0)*5.00/1024); //Conversion del valor leído entre 0 y 1024
                                   //lo pasamos a voltios

  lcd.print("V =>");
  lcd.setCursor(9,1); //Desplazamos el cursor sobre la segunda fila
  //Clasificamos según el voltaje detectado
  if ((analogRead(0)*5.00/1023.00) > 1.40)
  {
    lcd.print("Perfecto");
  }
  else if ((analogRead(0)*5.00/1023.00) > 1.20 && (analogRead(0)*5.00/1023.00) < 1.40)
  {
    lcd.print("Bueno");
  }
  else if ((analogRead(0)*5.00/1023.00) < 1.20 && (analogRead(0)*5.00/1023.00) > 0.30)
  {
    lcd.print("Regular");
  }
  else if ((analogRead(0)*5.00/1023.00) < 0.30)
  {
    lcd.print("Malo");
  }
  else
  {
    lcd.print("");
  }
  delay(250); //Esperamos para ver el efecto
}
```

Practica 4: Lectura de un sensor digital de temperatura.

4.1 Objetivos

- Conexión de un sensor con el Arduino

4.2 Material empleado.

- Arduino UNO
- Sensor temperatura
- Programación Labview.

4.3 Conocimientos Teóricos.

Una vez que tenemos conocimientos previos de programación con Arduino y sabemos cómo trabajar con el puerto serie, en esta práctica se pretende obtener la lectura de un sensor, bien sea, de temperatura, humedad o cualquier otro tipo. Vamos a conectar el sensor a la plataforma Arduino y mediante comunicación puerto serie vamos a enviar la información de la lectura al ordenador para un posterior procesado o con cualquier otro fin.

En esta práctica concretamente utilizaremos el sensor de temperatura mostrado en la Figura4.1 donde por un lado tenemos la caja del módulo de adecuación de señales y por otro lado el sensor o termopar.

El módulo de adecuación de señales posee un selector para variar el rango de temperaturas entorno al que nos movemos para tener una mayor precisión y por otro lado tenemos la salida de tensión bien de 0V a 1V o de -0.25V a 0.25V, por ejemplo si seleccionamos la escala de 0 a 100°C, la salida que obtendremos será para 0°C de 0V mientras que para los 100°C obtendremos 1V y para temperaturas intermedios obtendremos su nivel de tensión correspondiente. El valor de tensión será la salida que recogeremos con el Arduino para su posterior estudio y mediante una simple regla de tres podremos obtener el valor de la temperatura en cada instante.



Figura4.1

Y por otro lado tenemos la sonda que será la encargada de captar la temperatura a la cual la estemos sometiendo y que ira conectada al módulo.

4.4 Aplicación Practica

Una vez conocido el funcionamiento del termómetro en esta práctica como bien comentamos anteriormente se pretende realizar un programa que permita recoger los datos obtenidos por el sensor para un posterior procesado.

Practica 3: Modulación Por Ancho de Pulso.

3.1 Objetivos

- Conocer la finalidad de la modulación por ancho de pulso.
- Aplicación práctica de la modulación por ancho de pulso.

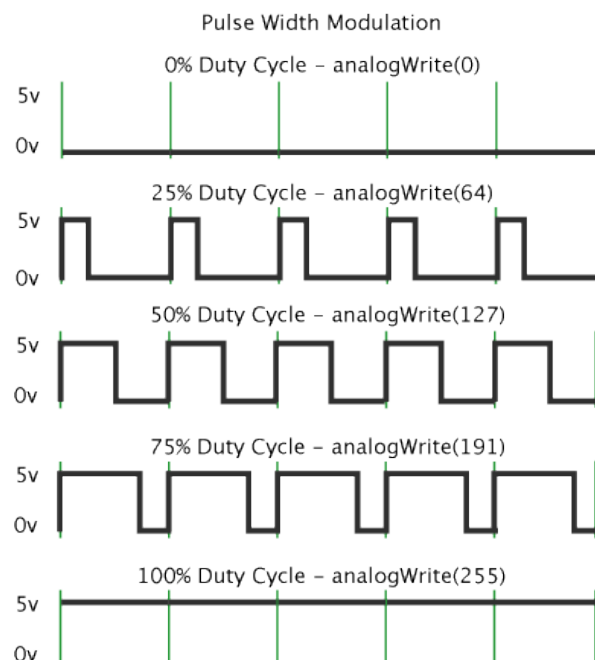
3.2 Material empleado.

- Arduino UNO.
- Diodo Led.

3.3 Conocimientos Teóricos.

La Modulación por Ancho de Pulso (PWM = Pulse Width Modulation) es una técnica para simular una salida analógica con una salida digital. El control digital se usa para crear una onda cuadrada, una señal que conmuta constantemente entre encendido y apagado. Este patrón de encendido-apagado puede simular voltajes entre 0 (siempre apagado) y 5 voltios (siempre encendido) simplemente variando la proporción de tiempo entre encendido y apagado. A la duración del tiempo de encendido (ON) se le llama Ancho de Pulso (pulse width). Para variar el valor analógico cambiamos, o modulamos, ese ancho de pulso. Si repetimos este patrón de encendido-apagado lo suficientemente rápido por ejemplo con un LED el resultado es como si la señal variara entre 0 y 5 voltios controlando el brillo del LED.

En el gráfico de abajo las líneas verticales representan un periodo regular. Esta duración o periodo es la inversa de la frecuencia del PWM. En otras palabras, con la Arduino la frecuencia PWM es bastante próxima a 500Hz lo que equivale a periodos de 2 milisegundos cada uno. La llamada a la función `analogWrite()` debe ser en la escala desde 0 a 255, siendo 255 el 100% de ciclo (siempre encendido), el valor 127 será el 50% del ciclo (la mitad del tiempo encendido), etc.



3.4 Conocimientos Prácticos: Ejemplo de aplicación.

Se pretende realizar mediante la modulación por ancho de pulso un programa que permita el control de la luminosidad de un led, es decir que empiece desde cero hasta el valor máximo de luminosidad y a continuación decremente. Además queremos visualizar el proceso de incremento y decremento a través del monitor del puerto serie.

Solución:

```
int ledPin1 = 9;    //Incializacion de las variables |
int ledPin2 = 10;   //correspondiente a cada pin

void setup() {
    Serial.begin(9600);
}

void loop() {
    // Mediante un for y una variable aleatoria hacemos el incremeto
    //para poder visualizar el apagado y encendido progresivo del led
    //Recordando que el valor de 0V corresponde con 0 y el de
    //5V con 255
    for(int fadeValue = 0 ; fadeValue <= 255; fadeValue +=5)
    { //vamos escribiendo el valor de tension que lleva esta variable
      //en el pin correspondiente a cada led
      analogWrite(ledPin1, fadeValue);
      analogWrite(ledPin2, fadeValue);
      Serial.println(fadeValue,DEC);
      // Esperamoso un tiempo para observar el efecto
      delay(30);
    }

    // Y de igual forma que incrementamos debemos de decrementar la
    //tension del led
    for(int fadeValue = 255 ; fadeValue >= 0; fadeValue -=5) {

        analogWrite(ledPin1, fadeValue);
        analogWrite(ledPin2, fadeValue);
        Serial.println(fadeValue,DEC);

        delay(30);
    }
}
```

Practica 2: Estructuras de Control

2.1 Objetivos.

- Conocimiento de las estructuras de control en el lenguaje de programación de Arduino.
- Manejo y utilización de las estructuras de control.
- Ejemplo de aplicación.

2.2 Material empleado.

- Arduino UNO.
- Diodo Led.

2.3 Conocimientos Teóricos.

A continuación se muestran todas las estructuras de control más usuales en la programación de Arduino.

- **If(condicional)**

Esta estructura puede ser usada en conjunto con uno o más operadores de comparación, comprueba si cierta condición se cumple, por ejemplo, si un input posee un valor mayor a cierto número. El formato para una comprobación *if* es el siguiente:

```
If(algunavariabile>50)
{
    // Realiza una operación
}
```

Veamos los distintos operadores de comparación:

```
x == y (x es igual a y)
x != y (x no es igual a y)
x < y (x es menor a y)
x > y (x es mayor a y)
x <= y (x es menor o igual a y)
x >= y (x es mayor o igual a y)
```

- **if... else.. o if...else if...**

if/else permite mayor control sobre el flujo del código que la declaración *if* básica, por permitir agrupar múltiples comprobaciones. Por ejemplo, una salida analógica podría ser comprobada, y tomarse una acción si el valor de la salida es menor a 500, y, otra acción si el valor es igual o mayor a 500. Ejm:

```
if (pinCincoInput < 500)
{ // acción A }
else
{ // acción B }
```


If/else if puede proceder a una comprobación **if**, de esta forma, se pueden realizar múltiples comprobaciones en una misma estructura de condiciones. Cada comprobación procederá a la siguiente, sólo cuando su propio resultado sea *FALSE*. Cuando el resultado sea *TRUE*, su bloque de código contenido, será ejecutado, y el programa esquivará las siguientes comprobaciones hasta el final de la estructura de comprobaciones. Si ninguna comprobación devuelve valor *TRUE*, el **else** será ejecutado, y de no haber ninguno declarado, simplemente no sucede nada.

Entonces un bloque **else if** puede ser usado con o sin **else** al final. La cantidad de declaraciones **else if**, y sus ramificaciones son ilimitadas. Ejm:

```
if (pinCincoInput < 500)
{
    // ejecutar A
}
else if (pinCincoInput >= 1000)
{
    // ejecutar B
}
else
{
    // ejecutar C
}
```

- **Switch/case**

Al igual que la sentencia **if**, **switch...case** controla el flujo de programas permitiendo a los programadores especificar diferentes códigos que deberían ser ejecutados en función de varias condiciones. En particular, una sentencia switch compara el valor de una variable con el valor especificado en las sentencias case. Cuando se encuentra una sentencia case cuyo valor coincide con dicha variable, el código de esa sentencia se ejecuta.

La palabra clave **break** sale de la sentencia switch, y es usada típicamente al final de cada case. Sin esta sentencia break, la estructura switch continuaría ejecutándose hasta encontrar un break o hasta llegar al final de la sentencia switch. Ejm:

```
switch (var) {
    case 1:
        //hacer algo cuando sea igual a 1
        break;
    case 2:
        //hacer algo cuando sea igual a 2
        break;
    default:
        // si nada coincide, ejecuta el "default"
        // el "default" es opcional
}
```

- **While**

Los bucles **while** se ejecutan continuamente, hasta que la expresión de dentro del paréntesis, (), pasa a ser falsa. Algo debe modificar la variable comprobada, si no el bucle **while** nunca terminará. Lo que modifique la variable puede estar en el código, como una variable que se incrementa, o ser una condición externa, como el valor que da un sensor. Ejm:

```
var = 0;
while(var < 200){
    // haz algo repetitivo 200 veces
    var++;
}
```

- **Do/While**

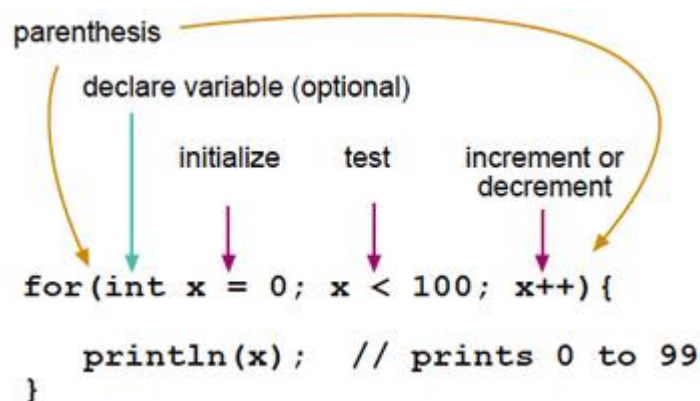
El bucle "do" trabaja de la misma manera que el bucle "while", con la excepción de que la condición se comprueba al final del bucle, por lo que este bucle se ejecuta "siempre" al menos una vez.

```
do
{
    delay(50); // espera a que los sensores se
    estabilicen
    x = readSensors(); // comprueba los sensores

} while(x < 100); //si se cumple la condición se repite
el bucle
```

- **For**

La declaración **for** es usada para repetir un bloque encerrado entre llaves. Un incremento de un contador es usado, normalmente, para aumentar y terminar con el bucle. La estructura **for** es muy útil para la mayoría de las operaciones repetitivas, y habitualmente se usa para operaciones con vectores, para operar sobre conjuntos de datos/pines



2.4 Ejemplo de Aplicación.

Se pretende realizar un programa que controle el tiempo de estado alto y bajo de un led, es decir que le introduzcamos por el puerto serie un numero de 0 a 9, que indican 0mseg y 900mseg de tiempo de estado alto y estado bajo, y podamos controlar el parpadeo del mismo. Por lo que vamos a programar con el fin de que nos pida por pantalla que introduzcamos el tiempo de estado alto y el tiempo de estado bajo y una vez introducido que se realice el control del parpadeo del led.

Notas: ASCII-48=DEC

Solución:

```
void setup() {
  Serial.begin(9600); //Inicializamos el puerto serie
  pinMode(13, OUTPUT); //Asignamos al pin 13 como salida donde conectaremos el LED
  //Escribimos por puerto serie la primera vez
  Serial.println("Introduce el tiempo de estado ALTO:");
}

//Inicializacion de variables
char datosbyte;
byte f=0;
int x=0;
int y=0;
void loop() {

  //Vamos a leer por puerto serie el valor de estado alto y bajo
  if (Serial.available() > 0) //Compruebo si el puerto serie esta accesible
  {
    if(f==0) //Utilizamos una variable llamada bandera que nos realizara
              //la funcion de conmutacion de tiempo alto y bajo
    {
      datosbyte=Serial.read(); //Leo los datos del puerto serie
      x=datosbyte-48; // paso de codigo ASCII a decimal
      Serial.print("El tiempo de estado ALTO es: ");
      Serial.println(x,DEC); // Muestro el tiempo introducido

      Serial.println("Introduce el tiempo de estado BAJ0:");
      f=1;
    }
  }

  if (Serial.available() > 0)
  {
    if(f==1)
    {
      datosbyte=Serial.read(); //Voi leyendo los datos del puerto serie
      y=datosbyte-48; //Convierto de codigo ASCII a decimal
      Serial.print("El tiempo de estado BAJ0 es: ");
      Serial.println(y,DEC); // Muestro el tiempo bajo introdcido

      Serial.println("Introduce el tiempo de estado ALTO:");
      f=0;
    }
  }

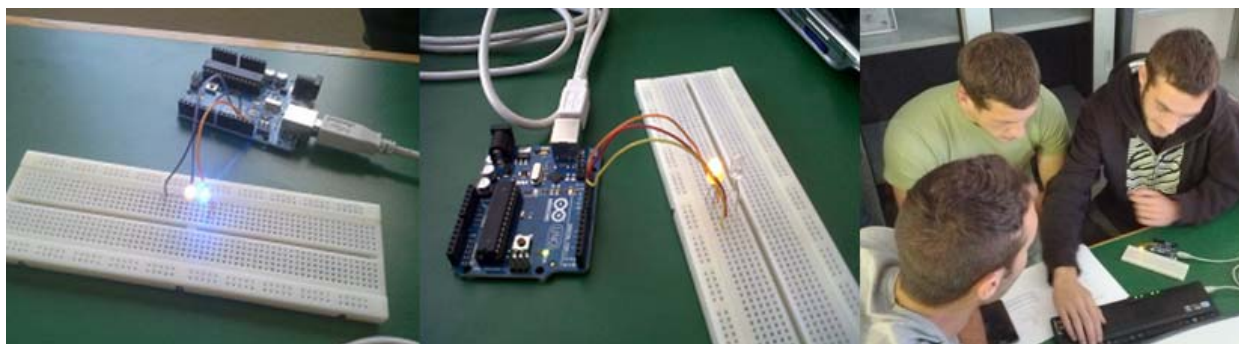
  digitalWrite(13, HIGH); // Pongo el Led en estado alto
  int x1=1000*x;           // multiplico los segundos introducidos para
  delay(x1);               // pasarlos a milisegundos
  digitalWrite(13, LOW);   // Pongo el Led en estado bajo
  int y1=1000*y;           // Vuelvo a multiplicar
  delay(y1);               // Y mantengo en estado bajo
}
```

Objetivo 1: Programación de micro-controladores, dedicados a medidas de diversas magnitudes físicas. Con esta actividad potenciaremos el uso de instrumentos de medida dedicados a una tarea concreta, muy presentes en instrumentos de medida de mano (portátiles) y en otros dispositivos portables.

Actividades realizadas y resultados obtenidos:

Para la consecución de este objetivo ha sido necesaria la adquisición y uso de la placa micro-controlada programable ARDUINO como herramienta didáctica que, gracias a su versatilidad y fácil instalación, está experimentando una amplia difusión en la docencia electrónica. Concretamente se han comprado 8 placas básicas ARDUINO UNO (apropiadas para la iniciación en la programación de micro-controladores y con capacidad para el desarrollo de aplicaciones de nivel intermedio), y 2 placas de un nivel superior ARDUINO MEGA (que permiten abordar proyectos de mayor entidad para brindar posibilidades a la hora de realizar Proyectos Fin de Carrera y Trabajos Fin de Máster).

- Durante el curso 2011/2012, dentro de las prácticas de laboratorio de la asignatura Electrónica en el Grado de Ingeniería en Tecnologías Industriales, se ha desarrollado una práctica con microcontroladores ARDUINO pensada como una primera toma de contacto con dispositivos de electrónica avanzada –en este punto del grado los alumnos sólo han trabajado con elementos electrónicos básicos como resistencias, condensadores o diodos. La práctica consiste en el control de LEDs mediante la programación de la placa de manera que el alumno debe conseguir ciertas secuencias de iluminación problema. Para conseguir estas secuencias se deben usar tanto las salidas analógicas como digitales de la placa. Así, con esta práctica el alumno adquiere un conocimiento básico sobre la programación y conexión de la placa que será muy útil cuando se vuelva a reincidir sobre este dispositivo en cursos posteriores. Los resultados han sido muy satisfactorios teniendo en cuenta el interés que han mostrado los alumnos por ARDUINO. De hecho nos comentaban que era la primera vez que veían como un programa repercutía en un elemento físico externo al ordenador y pedían información sobre el precio y la forma de adquirir estas placas.



Imágenes extraídas de las memorias de prácticas de los alumnos de Electrónica de Grado.

- En la asignatura Instrumentación Electrónica de la titulación Ingeniería Técnica Industrial en Electrónica Industrial (curso 2011/2012) se ha realizado Cinco prácticas ARDUINO desde iniciación a nivel intermedio, en el control y gestión de entradas y salidas analógicas y digitales y aplicaciones de medida (Se incluyen como anexos). De nuevo los resultados han sido satisfactorios en función de la disposición de los alumnos a trabajar con este tipo de esquemas donde dispositivos físicos e informática interaccionan. Dado que una parte importante de la práctica consiste en el desarrollo de paneles con LabVIEW este punto será de nuevo comentado y ampliado en el Objetivo 2.
- Con el fin de mostrar las posibilidades de ARDUINO, y dado que las prácticas realizadas durante este curso pueden considerarse de iniciación, creímos oportuno desarrollar un dispositivo de nivel avanzado y hacer una demostración para concluir las prácticas anteriormente descritas. El dispositivo en cuestión era un autómata equipado con unos sensores lumínicos direccionales. Estos sensores se conectaban a las entradas de la placa de manera que ésta podía discernir qué sensor estaba más iluminado y por tanto la dirección de la provenía la mayor cantidad de luz. Procesando esta información la placa activaba unos motores acoplados a ruedas para girar y avanzar hacia la dirección en que provenía la luz desarrollando así un comportamiento “inteligente” (ver http://www.youtube.com/watch?v=4cedJHsK_Hc). Después de haber trabajado durante la práctica con ARDUINO, y tras una somera explicación de los componentes y la programación empleada en el autómata, los alumnos eran capaces de entender el funcionamiento e interacción de elementos y por tanto el modo en

que los sistemas electrónicos avanzados reciben, procesan y responden de manera compleja a la información física exterior gracias al uso de microprocesadores. De nuevo el nivel de interés de los alumnos fue muy elevado, mostrando algunos de ellos su predisposición a realizar proyectos futuros semejantes y si éstos justificarían un proyecto fin de carrera o fin de master.

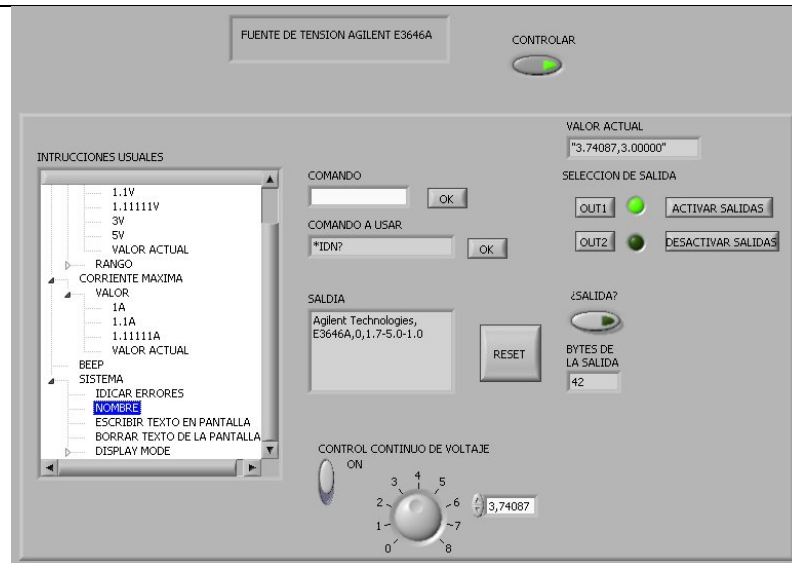


Objetivo 2: Diseño de paneles de instrumentación virtual mediante LabVIEW, que es la referencia mundial en esta materia. Con esta actividad se pretende potenciar la programación en LabVIEW para los títulos, y que hemos aprendido como investigadores en nuestro grupo PAIDI-TIC-168 en Instrumentación Computacional y Electrónica Industrial (ICEI).

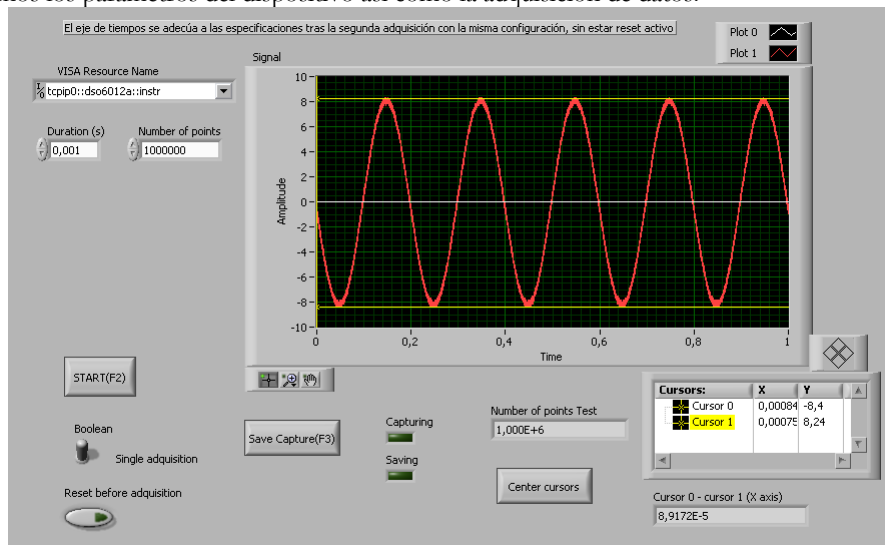
Actividades realizadas y resultados obtenidos:

Este objetivo ha tenido durante este curso su reflejo en una asignatura, mientras que se han sentado las bases para la utilización en otras asignaturas en los próximos cursos.

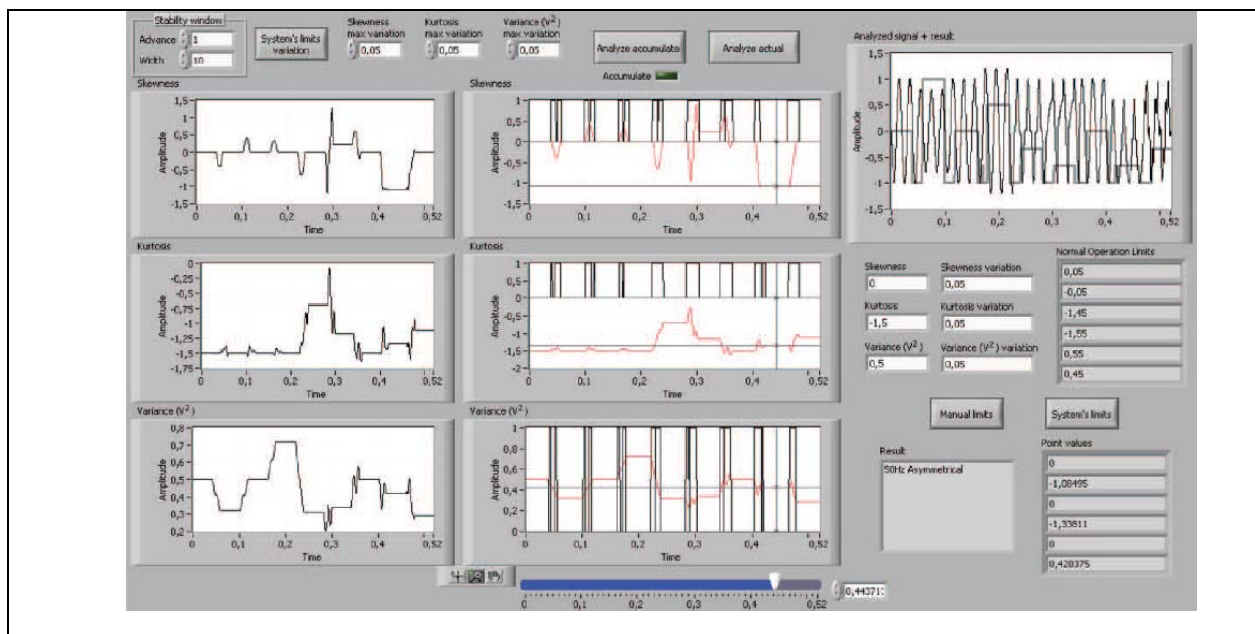
- **Práctica de monitorización de Temperaturas** en Instrumentación Electrónica (I.T.I. E.I.). consistente en la conexión de sensores de temperatura PhilipHarris E60170/1 a través de las entradas analógicas de la placa de manera que los registros son almacenados en la memoria de ésta y enviados a través de puerto serie al ordenador, donde una interfaz de LabVIEW procesa los datos y los representa en un panel diseñado para ello.
- **Control de fuente de CC Agilent E2646A.** Este desarrollo se usará en cursos próximos como ejemplo de control de sistemas a través de puertos GPIB. Para ello se ha desarrollado un módulo en LabVIEW con un panel que controla todos los parámetros del dispositivo (voltaje, intensidad máxima, display,...) cuyo resultado es la posibilidad de obtener señales variables en el tiempo a partir de una fuente continua. Así los alumnos trabajan con conexionado, programación LabVIEW, dispositivos electrónicos y pueden testear el resultado total de la implementación a través de la señal de salida.



- Control del osciloscopio Agilent DSO6012A.** Este desarrollo se usará en cursos próximos como ejemplo de control de sistemas a través de tarjetas Ethernet. Esto permite el acceso vía LAN al osciloscopio desde cualquier punto Escuela Politécnica Superior de Algeciras. De nuevo, mediante una interfaz LabVIEW controlamos los parámetros del dispositivo así como la adquisición de datos.



- **Instrumento para la medición de la calidad de la energía eléctrica.** Nuestro grupo de Investigación PAIDI-TIC-168 en Instrumentación Computacional y Electrónica Industrial (ICEI) tiene concedido actualmente un proyecto nacional que trata sobre la monitorización de la calidad en el suministro de energía eléctrica TEC2010-19242-C03-03 (SIDER-HOSAPQ). Hemos creído interesante implementar los resultados de este proyecto en paneles LabVIEW. De este modo, y al igual que ocurría con el autómatas en el Objetivo 1, los alumnos pueden observar/manejar/testear una aplicación de nivel avanzado en LabVIEW, completando y reforzando el conocimiento adquirido en las actividades planteadas con esta herramienta.



Objetivo 3: Integración de instrumentos de medida en internet y control de los mismos desde cualquier ordenador de la universidad, con extensibilidad hacia fuera de la universidad. Se pretende iniciar una vía formativa práctica en este ámbito.

Actividades realizadas y resultados obtenidos:

- **Control del osciloscopio Agilent DSO6012A.** Como se ha comentado en el Objetivo 2, el control de este osciloscopio se hace vía LAN. Gracias a una dirección IP propia suministrada por el CITI, el alumno puede comprobar como desde cualquier punto de la red local puede tener acceso al control básico del osciloscopio gracias a la interfaz Web que proporciona el fabricante; o bien, usando la interfaz LabVIEW comentada en el Objetivo 2 puede tener acceso al control y a la adquisición de datos.
- **Sistema de alarmas vía Internet:** Dada la compatibilidad que ofrece LabVIEW con .NET framework estamos desarrollando aplicaciones que envíen mensajes de correo electrónico ante ciertos eventos. Actualmente, hemos desarrollado un sistema de alarmas para el **Instrumento para la medición de la calidad de la energía eléctrica** (ver Objetivo 2) que se activa cuando el sistema se ha bloqueado. Se pretende, utilizando los mismos elementos, dotar de sistemas de alarma y de transmisión de datos a las aplicaciones LabVIEW propuestas a alumnos.

Objetivo 4: Diseño de aplicaciones informáticas para la docencia de circuitos electrónicos

Actividades realizadas y resultados obtenidos:

Se han realizado una serie de simulaciones sobre contenidos clave en la docencia de las asignaturas (Electrónica y Electrónica de potencia (G.I.T.I.)). Gracias a ellos el alumno podrá trabajar/estudiar/comprobar el comportamiento de elementos, circuitos o propiedades electrónicas cambiando parámetros característicos. Del mismo modo, podrá usar estas aplicaciones como generador de problemas resueltos adaptados a la nomenclatura y metodologías usados en las asignaturas concernidas. Se ha desarrollado en OpenOffice para facilitar su difusión.

- **Electrónica de Grado**
Bandas de Energía
Concentración – Cálculos de concentración de átomos para distintos elementos
Conducción – Se calculan parámetros de conducción en metales
Nivel de Fermi – De distintos metales

Semiconductores y unión PN

Semiconductores I – Concentraciones, velocidades de portadores mayoritarios y minoritarios y tipología

Semiconductores II – Cálculo de la distancia del Nivel de Fermi

Efecto Hall – Conductividad y movilidad de portadores

Diodo – Resolución de circuito típico Diodo y Resistencia en Serie (Gráficas: Recta de carga del circuito y V-I del diodo)

Transistor JBT

Polarización fija – Resuelve el circuito indicando la región de trabajo del transistor (Gráficas: Rectas de carga en activa)

Colector-Base –

Autopolarizado-

Transistor JFET

Análisis JFET – Definiendo el estado de los terminales, determina la región de trabajo y la corriente que circula por el drenador

Análisis MOSFET deplexión –

Análisis MOSFET acumulación –

Dos NMOS serie – Problema típico.

- **Electrónica de potencia**

Circuitos básicos de electrónica

RC carga

RLC paralelo

RLC serie

Convertidores CC-CC

Elevador

Reductor

Inverso

Rectificadores Monofásicos Controlados

Media onda con carga resistiva

Onda completa con carga resistiva

Onda completa con carga R-L

Objetivo 5: Diseño de actividades de evaluación a distancia que consistan en tomar medidas y programar los instrumentos vía internet, programar micro-controladores a partir de una batería de ejercicios con un marcado carácter práctico.

Actividades realizadas y resultados obtenidos:

Para la consecución de este objetivo es necesaria la completa funcionalidad de los elementos descritos en los Objetivos anteriores. Por ello el desarrollo total de este punto se plantea a más largo plazo. Avances necesarios en este sentido son:

- Programación de microcontroladores (Objetivo 1)
- Control del osciloscopio Agilent DSO6012A (Objetivo 2 y Objetivo 3)
- Sistema de alarmas vía Internet (Objetivo 3)

HC Serial Bluetooth Products

User Instructional Manual

1 Introduction

HC serial Bluetooth products consist of Bluetooth serial interface module and Bluetooth adapter, such as:

(1) Bluetooth serial interface module:

Industrial level: HC-03, HC-04(HC-04-M, HC-04-S)

Civil level: HC-05, HC-06(HC-06-M, HC-06-S)
HC-05-D, HC-06-D (with baseboard, for test and evaluation)

(2) Bluetooth adapter:

HC-M4

HC-M6

This document mainly introduces Bluetooth serial module. Bluetooth serial module is used for converting serial port to Bluetooth. These modules have two modes: master and slaver device. The device named after even number is defined to be master or slaver when out of factory and can't be changed to the other mode. But for the device named after odd number, users can set the work mode (master or slaver) of the device by AT commands.

HC-04 specifically includes:

Master device: HC-04-M, M=master

Slave device: HC-04-S, S=slaver

The default situation of HC-04 is slave mode. If you need master mode, please state it clearly or place an order for HC-04-M directly. The naming rule of HC-06 is same.

When HC-03 and HC-05 are out of factory, one part of parameters are set for activating the device. The work mode is not set, since user can set the mode of HC-03, HC-05 as they want.

The main function of Bluetooth serial module is replacing the serial port line, such as:

1. There are two MCUs want to communicate with each other. One connects to Bluetooth master device while the other one connects to slave device. Their connection can be built once the pair is made. This Bluetooth connection is equivalently liked to a serial port line connection including RXD, TXD

signals. And they can use the Bluetooth serial module to communicate with each other.

2. When MCU has Bluetooth slave module, it can communicate with Bluetooth adapter of computers and smart phones. Then there is a virtual communicable serial port line between MCU and computer or smart phone.

3. The Bluetooth devices in the market mostly are slave devices, such as Bluetooth printer, Bluetooth GPS. So, we can use master module to make pair and communicate with them.

Bluetooth Serial module's operation doesn't need drive, and can communicate with the other Bluetooth device who has the serial. But communication between two Bluetooth modules requires at least two conditions:

(1) The communication must be between master and slave.

(2) The password must be correct.

However, the two conditions are not sufficient conditions. There are also some other conditions basing on different device model. Detailed information is provided in the following chapters.

In the following chapters, we will repeatedly refer to Linvor's (Formerly known as Guangzhou HC Information Technology Co., Ltd.) material and photos.

2 Selection of the Module

The Bluetooth serial module named even number is compatible with each other; The slave module is also compatible with each other. In other word, the function of HC-04 and HC-06, HC-03 and HC-05 are mutually compatible with each other. HC-04 and HC-06 are former version that user can't reset the work mode (master or slave). And only a few AT commands and functions can be used, like reset the name of Bluetooth (only the slaver), reset the password, reset the baud rate and check the version number. The command set of HC-03 and HC-05 are more flexible than HC-04 and HC-06's. Generally, the Bluetooth of HC-03/HC-05 is recommended for the user.

Here are the main factory parameters of HC-05 and HC-06. Pay attention to the differences:

HC-05	HC-06
Master and slave mode can be switched	Master and slave mode can't be switched
Bluetooth name: HC-05	Bluetooth name: linvor
Password:1234	Password:1234

<p>Master role: have no function to remember the last paired slave device. It can be made paired to any slave device. In other words, just set AT+CMODE=1 when out of factory. If you want HC-05 to remember the last paired slave device address like HC-06, you can set AT+CMODE=0 after paired with the other device. Please refer the command set of HC-05 for the details.</p>	<p>Master role: have paired memory to remember last slave device and only make pair with that device unless KEY (PIN26) is triggered by high level. The default connected PIN26 is low level.</p>
<p>Pairing: The master device can not only make pair with the specified Bluetooth address, like cell-phone, computer adapter, slave device, but also can search and make pair with the slave device automatically.</p> <p>Typical method: On some specific conditions, master device and slave device can make pair with each other automatically. (This is the default method.)</p>	<p>Pairing: Master device search and make pair with the slave device automatically.</p> <p>Typical method: On some specific conditions, master and slave device can make pair with each other automatically.</p>
<p>Multi-device communication: There is only point to point communication for modules, but the adapter can communicate with multi-modules.</p>	<p>Multi-device communication: There is only point to point communication for modules, but the adapter can communicate with multi-modules.</p>
<p>AT Mode 1: After power on, it can enter the AT mode by triggering PIN34 with high level. Then the baud rate for setting AT command is equal to the baud rate in communication, for example: 9600.</p> <p>AT mode 2: First set the PIN34 as high level, or while on powering the module set the PIN34 to be high level, the Baud rate used here is 38400 bps.</p> <p>Notice: All AT commands can be operated only</p>	<p>AT Mode: Before paired, it is at the AT mode. After paired it's at transparent communication.</p>

<p>when the PIN34 is at high level. Only part of the AT commands can be used if PIN34 doesn't keep the high level after entering to the AT mode. Through this kind of designing, set permissions for the module is left to the user's external control circuit, that makes the application of HC-05 is very flexible.</p>	
<p>During the process of communication, the module can enter to AT mode by setting PIN34 to be high level. By releasing PIN34, the module can go back to communication mode in which user can inquire some information dynamically. For example, to inquire the pairing is finished or not.</p>	<p>During the communication mode, the module can't enter to the AT mode.</p>
<p>Default communication baud rate: 9600, 4800-1.3M are settable.</p>	<p>Default communication baud rate: 9600, 1200-1.3M are settable.</p>
<p>KEY: PIN34, for entering to the AT mode.</p>	<p>KEY: PIN26, for master abandons memory.</p>
<p>LED1: PIN31, indicator of Bluetooth mode. Slow flicker (1Hz) represents entering to the AT mode2, while fast flicker(2Hz) represents entering to the AT mode1 or during the communication pairing. Double flicker per second represents pairing is finished, the module is communicable.</p> <p>LED2: PIN32, before pairing is at low level, after the pairing is at high level.</p> <p>The using method of master and slaver's indicator is the same.</p> <p>Notice: The PIN of LED1 and LED2 are connected with LED+.</p>	<p>LED: The flicker frequency of slave device is 102ms. If master device already has the memory of slave device, the flicker frequency during the pairing is 110ms/s. If not, or master has emptied the memory, then the flicker frequency is 750m/s. After pairing, no matter it's a master or slave device, the LED PIN is at high level.</p> <p>Notice: The LED PIN connects to LED+ PIN.</p>
<p>Consumption: During the pairing, the current is</p>	<p>Consumption: During the pairing, the current is</p>

fluctuant in the range of 30-40mA. The mean current is about 25mA. After paring, no matter processing communication or not, the current is 8mA. There is no sleep mode. This parameter is same for all the Bluetooth modules.	fluctuant in the range of 30-40 m. The mean current is about 25mA. After paring, no matter processing communication or not, the current is 8mA. There is no sleep mode. This parameter is same for all the Bluetooth modules.
Reset: PIN11, active if it's input low level. It can be suspended in using.	Reset: PIN11, active if it's input low level. It can be suspended in using.
Level: Civil	Level: Civil

The table above that includes main parameters of two serial modules is a reference for user selection.

HC-03/HC-05 serial product is recommended.

3. Information of Package

The PIN definitions of HC-03, HC-04, HC-05 and HC-06 are kind of different, but the package size is the same: 28mm * 15mm * 2.35mm.

The following figure 1 is a picture of HC-06 and its main PINs. Figure 2 is a picture of HC-05 and its main PINs. Figure 3 is a comparative picture with one coin. Figure 4 is their package size information. When user designs the circuit, you can visit the website of Guangzhou HC Information Technology Co., Ltd. (www.wavesen.com) to download the package library of protle version.

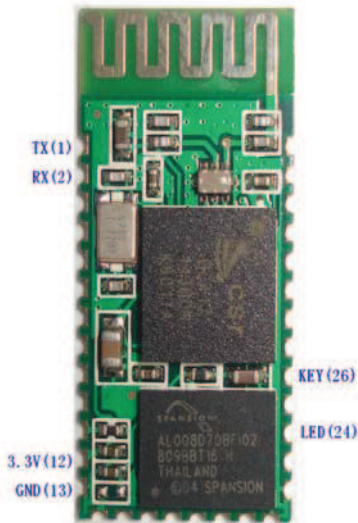


Figure 1 HC-06

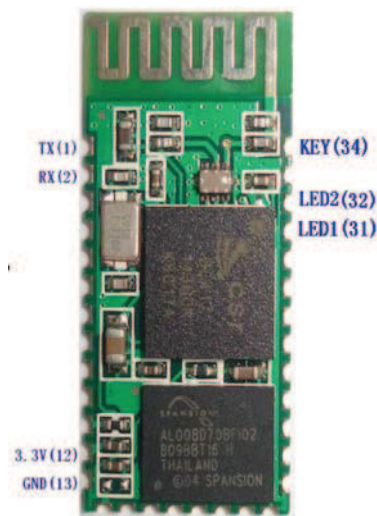


Figure 2 HC-05

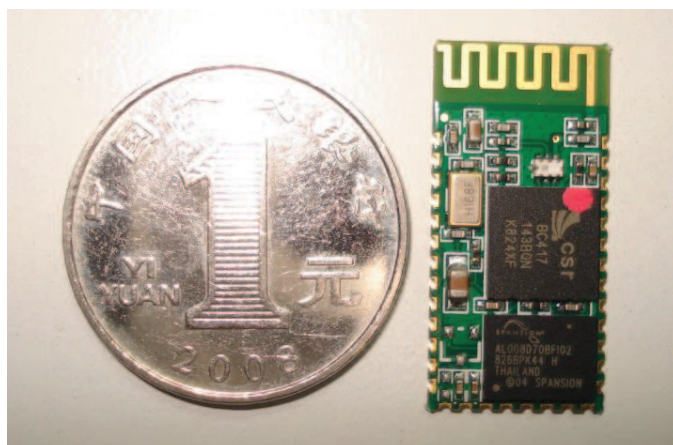
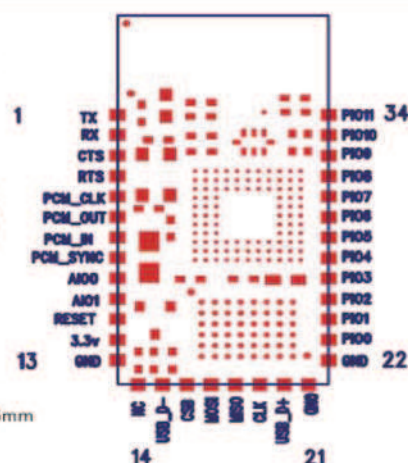
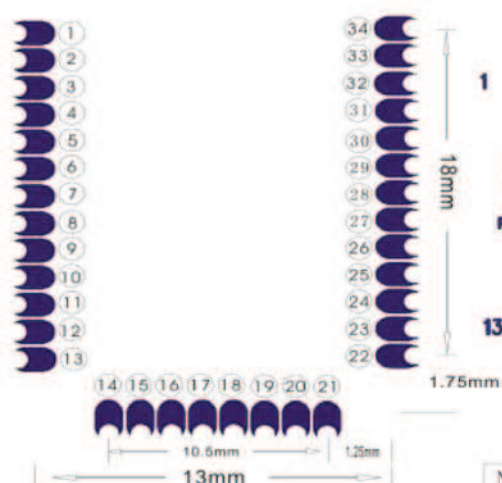


Figure 3 Comparative picture with one coin

LINOR BLUE T
www.linor.com

LV-BC-2.0

单位: mm



NO	PIN NAME	NO	PIN NAME
1	TX	20	USB D+
2	RX	21	GND
3	CTS	22	GND
4	RTS	23	PI00
5	PCM CLK	24	PI01
6	PCM OUT	25	PI02
7	PCM IN	26	PI03
8	PCM SYNC	27	PI04
9	AI00	28	PI05
10	AI01	29	PI06
11	RESET	30	PI07
12	3.3V	31	PI08
13	GND	32	PI09
14	NC	33	PI010
15	USB D-	34	PI011
16	CSB		
17	MOSI		
18	MISO		
19	CLK		



Figure 4 Package size information

4. The Using and Testing Method of HC-06 for the First Time

This chapter will introduce the using method of HC-06 in detail. User can test the module according to this chapter when he or she uses the module at the first time.

PINs description:

PIN1	UART_TXD , TTL/CMOS level, UART Data output
PIN2	UART_RXD, TTL/COMS level, s UART Data input
PIN11	RESET, the reset PIN of module, inputting low level can reset the module, when the module is in using, this PIN can connect to air.
PIN12	VCC, voltage supply for logic, the standard voltage is 3.3V, and can work at 3.0-4.2V
PIN13	GND
PIN22	GND
PIN24	LED, working mode indicator Slave device: Before paired, this PIN outputs the period of 102ms square wave. After paired, this PIN outputs high level. Master device: On the condition of having no memory of pairing with a slave device, this PIN outputs the period of 110ms square wave. On the condition of having the memory of pairing with a slave device, this PIN outputs the period of 750ms square wave. After paired, this PIN outputs high level.
PIN26	For master device, this PIN is used for emptying information about pairing. After emptying, master device will search slaver randomly, then remember the address of the new got slave device. In the next power on, master device will only search this address.

(1) The circuit 1 (connect the module to 3.3V serial port of MCU) is showed by figure 5.

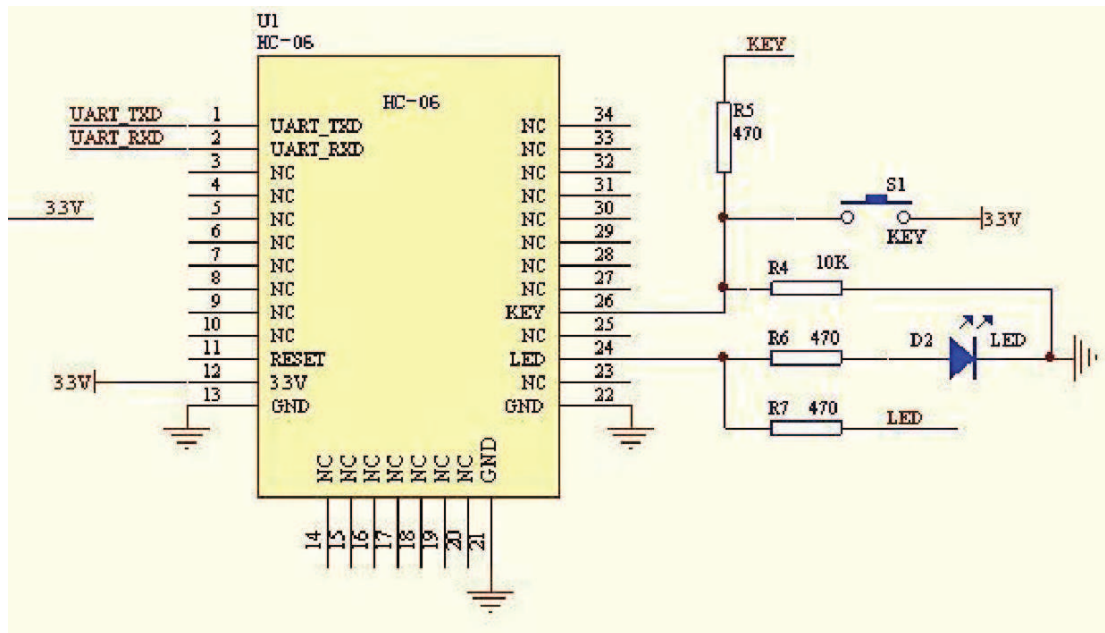


Figure 5 The circuit 1

In principle, HC-06 can work when UART_TXD, UART_RXD, VCC and GND are connected. However, for better testing results, connecting LED and KEY are recommended (when testing the master).

Where, the 3.3V TXD of MCU connects to HC-06's UART_RXD, the 3.3V RXD of MCU connects to HC-06's UART_TXD, and 3.3V power and GND should be connected. Then the minimum system is finished.

Note that, the PIN2:UART_RXD of Bluetooth module has no pull-up resistor. If the MCU TXD doesn't have pull-up function, then user should add a pull-up resistor to the UART_RXD. It may be easy to be ignored.

If there are two MCU which connect to master and slave device respectively, then before paired(LED will flicker) user can send AT commands by serial port when the system is power on. Please refer to HC-04 and HC-06's data sheet for detailed commands. In the last chapter, the command set will be introduced. Please pay attention to that the command of HC-04/HC-06 doesn't have terminator. For example, consider the call command, sending out AT is already enough, need not add the CRLF (carriage return line feed).

If the LED is constant lighting, it indicates the pairing is finished. The two MCUs can communicate with each other by serial port. User can think there is a serial port line between two MCUs.

(2) The circuit 2 (connect the module to 5V serial port of MCU) is showed by figure 6.

Figure 6 is the block diagram of Bluetooth baseboard. This kind of circuit can amplify Bluetooth module's operating voltage to 3.1-6.5V. In this diagram, the J1 port can not only be connected with MCU system of 3.3V and 5V, but also can be connected with computer serial port.

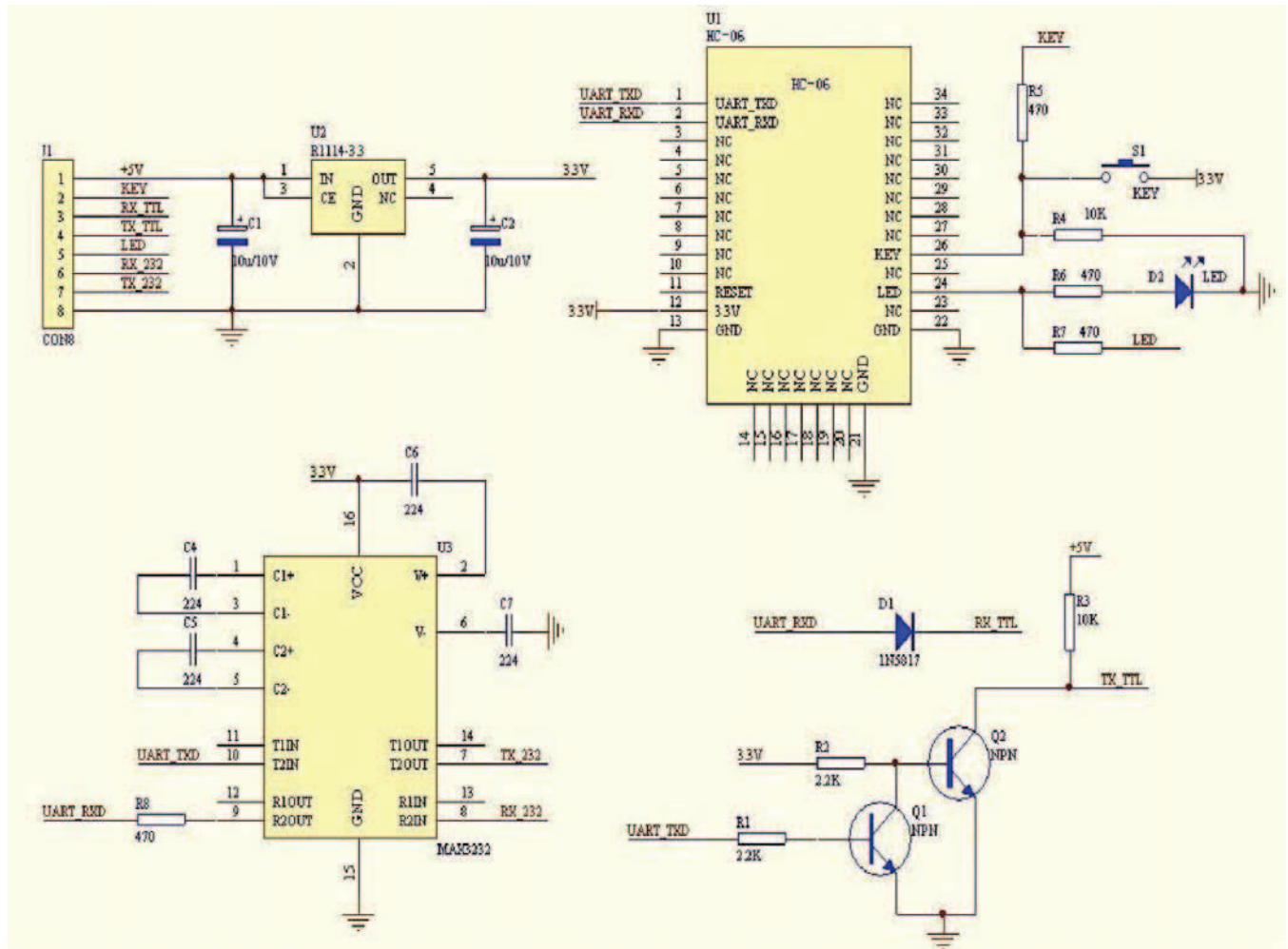


Figure 6 The circuit 2

(3) AT command test

Before paired, the mode of HC-04 and HC-06 are AT mode.

On the condition of 9600N81, OK will be received when user send the two letters AT. Please refer to the last chapter of datasheet for other commands of HC-06. Please pay attention to that sending out AT is already enough, need not add the CRLF (carriage return line feed).

The command set of Version V1.4 doesn't include parity. The version V1.5 and its later version have parity function. Moreover, there are three more commands of V1.5 than V1.4. They are:

No parity (default) AT+PN

Odd parity	AT+PO
Even parity	AT+PE

Do not let the sending frequency of AT command of HC-06 exceed 1Hz, because the command of HC-06 end or not is determined by the time interval.

(4) Pairing with adapter

User can refer to the download center of the company's website for "The Introduction of IVT" that introduces the Bluetooth module makes pair with computer adapter. That document taking HC-06-D for example introduces how the serial module makes pair with the adapter. That method is like to make pair with cell-phone. But the difference is that cell-phone need a third-party communication software to help. It's liked the kind of PC serial helper of and the hyper terminal. A software named "PDA serial helper" provided by our company is suitable for WM system. It has been proven that this serial module is supported by many smart phone systems' Bluetooth, such as, sybian, android, windows mobile and etc.

(5) Pairing introduction

HC-06 master device has no memory before the first use. If the password is correct, the mater device will make pair with the slave device automatically in the first use. In the following use, the master device will remember the Bluetooth address of the last paired device and search it. The searching won't stop until the device is found. If master device's PIN26 is input high level, the device will lose the memory. In that occasion, it'll search the proper slave device like the first use. Based on this function, the master device can be set to make pair with the specified address or any address by user.

(6) Reset new password introduction

User can set a new password for the HC-06 through AT+PINxxxx command. But the new password will become active after discharged all the energy of the module. If the module still has any energy, the old one is still active. In the test, for discharging all the system energy and activating the new password, we can connect the power supply PIN with GND about 20 seconds after the power is cut off. Generally, shutting down the device for 30 minutes also can discharge the energy, if there is no peripheral circuit helps discharge energy. User should make the proper way according to the specific situation.

(7) Name introduction

If the device has no name, it's better that user doesn't try to change the master device name. The name should be limited in 20 characters.

Summary: The character of HC-06: 1 not many command 2 easy for application 3 low price. It's good for some specific application. HC-04 is very similar with HC-06. Their only one difference is HC-04 is for industry, HC-06 is for civil. Except this, they don't have difference.

The following reference about HC-04 and HC-06 can be downloaded from company website

www.wavesen.com:

HC-06 datasheet .pdf	(the command set introduction is included)
HC-04 datasheet .pdf	(the command set introduction is included)
IVT BlueSoleil-2.6	(IVT Bluetooth drive test version)
Bluetooth FAQ.pdf	
HC-04-D(HD-06-D)datasheet(English).pdf	
HC-06-AT command software (test version)	(some commands in V1.5 is not supported by V1.4)
PCB package of Bluetooth key modules	(PCB package lib in protel)
IVT software manual.pdf	(introduce how to operate the module and make pair with Bluetooth module)
PDA serial test helper.exe	(serial helper used for WM system)

5 manual for the first use of HC-05

This chapter will introduce how to test and use the HC-05 if it's the first time for user to operate it.

(1) PINs description

PIN1	UART_TXD, Bluetooth serial signal sending PIN, can connect with MCU's RXD PIN
PIN2	UART_RXD, Bluetooth serial signal receiving PIN, can connect with the MCU's TXD PIN, there is no pull-up resistor in this PIN. But It needs to be added an external pull-up resistor.
PIN11	RESET, the reset PIN of module, inputting low level can reset the module, when the module is in using, this PIN can connect to GND.
PIN12	VCC, voltage supply for logic, the standard voltage is 3.3V, and can work at 3.0-4.2V
PIN13	GND

(3) Application circuit 2 (connect to 5V serial system or PC serial)

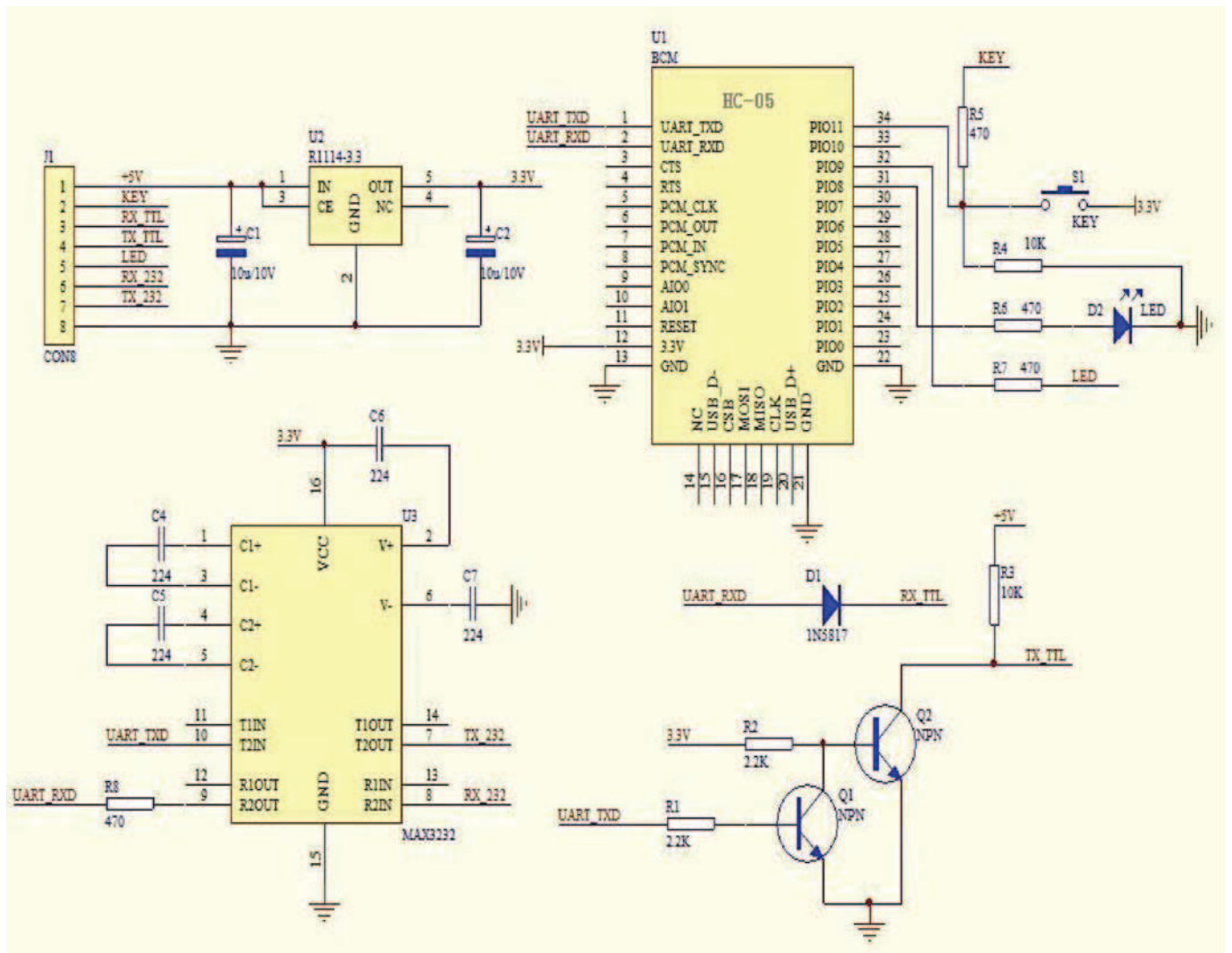


Figure 8 Application circuit 2

(4) AT command test

This chapter introduces some common commands in use. The detail introduction about HC-05 command is in HC-0305 AT command set.

Enter to AT mode:

Way1: Supply power to module and input high level to PIN34 at the same time, the module will enter to AT mode with the baud rate-38400.

Way2: In the first step, supply power to module; In the second step, input high level to PIN34. Then the module will enter to AT mode with the baud rate-9600. Way1 is recommended.

Command structure: all command should end up with “\r\n” (Hex: 0X0D X0A) as the terminator. If

the serial helper is installed, user just need enter “ENTER” key at the end of command.

Reset the master-slave role command:

AT+ROLE=0 ----Set the module to be slave mode. The default mode is slave.

AT+ROLE=1 ----Set the module to be master mode.

Set memory command:

AT+CMODE=1

Set the module to make pair with the other random Bluetooth module (Not specified address). The default is this mode.

AT+CMODE=1

Set the module to make pair with the other Bluetooth module (specified address). If set the module to make pair with random one first, then set the module to make pair with the Bluetooth module has specified address. Then the module will search the last paired module until the module is found.

Reset the password command

AT+PSWD=XXXX

Set the module pair password. The password must be 4-bits.

Reset the baud rate

AT+UART== <Param>,<Param2>,<Param3>.

More information is provided at HC-0305 command set

Example:

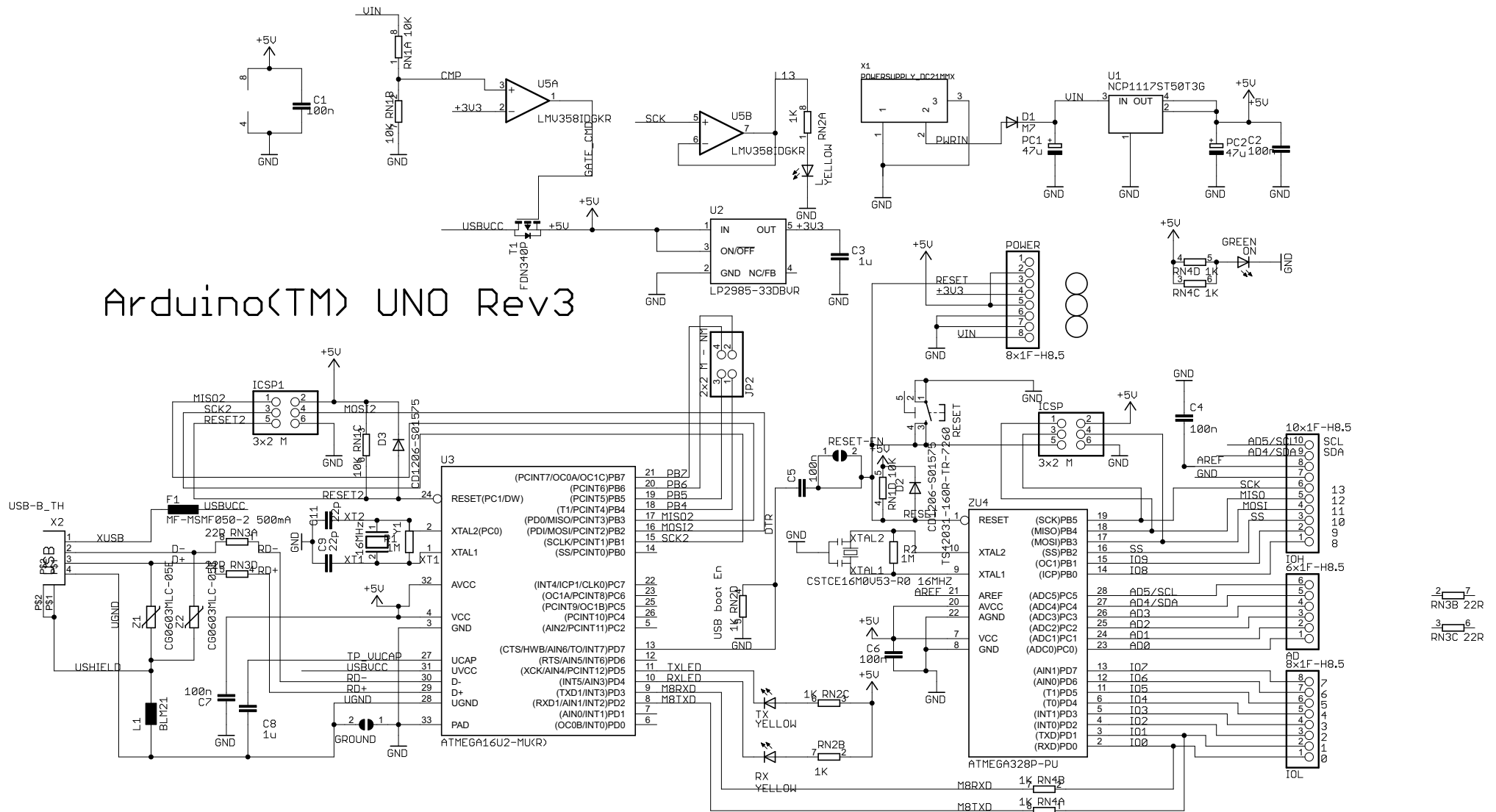
AT+UART=9600,0,0 ----set the baud rate to be 9600N81

Reset the Bluetooth name

AT+NAME=XXXXX

Summary:

HC-05 has many functions and covers all functions of HC-06. The above commands are the most common ones. Besides this, HC-05 leaves lots of space for user. So HC-05 is better than HC-06 and



Reference Designs ARE PROVIDED "AS IS" AND "WITH ALL FAULTS. Arduino DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, REGARDING PRODUCTS, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Arduino may make changes to specifications and product descriptions at any time, without notice. The Customer must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Arduino reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The product information on the Web Site or Materials is subject to change without notice. Do not finalize a design with this information.

ARDUINO is a registered trademark.

Use of the ARDUINO name must be compliant with <http://www.arduino.cc/en/Main/Policy>

SENSOR ULTRASONIDO

Los sensores ultrasónicos usan sonar para determinar la distancia de un objeto como los murciélagos o delfines lo hacen, ofrecen un excelente rango de detección sin contacto, su operación no es afectada por la luz del sol o materiales oscuros, aunque materiales acústicamente blandos son difíciles de detectar.

Características:

Voltaje de alimentación: +5VDC

Corriente en espera: <2mA

Corriente de trabajo: 15mA

Ángulo eficaz: <15°

Rangos de distancia: 2cm a 400cm ó 1" a 13ft

Resolución: 0.3cm

Ángulo de medida: 30°

Ancho de pulso de disparo (Trigger Input Pulse Width): 10us se aplica pulso de (10us en Trig)

Eco (Echo): salida del sensor.

Frecuencia de ultrasonido: 40KHz

PINES DE CONEXIÓN



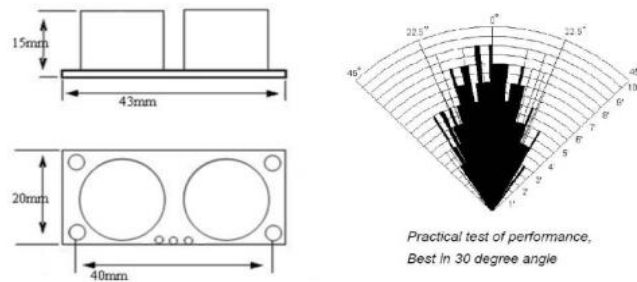
VCC = +5VDC

Trig = Trigger input of Sensor, entrada del pulso de disparo para iniciar la medición.

Echo = Echo output of Sensor, pin de eco, que retorna un pulso proporcional a la distancia que rebota el sonido.

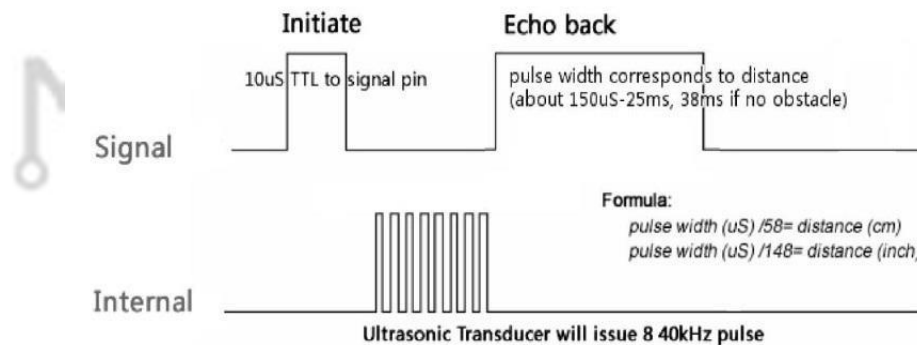
GND = GND

Dimensiones y ángulos de desempeño:



Operación

Para iniciar la medición el pin **Trig**, el de la señal de disparo (trigger) debe recibir un pulso de cinco voltios por al menos diez micro segundos, esto indicará al sensor que transmita una ráfaga de ocho ciclos ultrasónicos a 40KHz y espere por la ráfaga reflejada cuando el sensor detecta la señal de ultrasonido en el receptor enviará una señal en alto (5v) por el pin **Echo**, esta señal tendrá un retardo (ancho) proporcional a la distancia, así que para obtener la distancia se debe medir el Ton (tiempo en alto en el pin Echo)



Tiempo que tarda la señal en volver es igual al ancho del pulso en el pin Echo en microsegundos, para determinar la distancia a partir de este tiempo:

$$\text{Distancia en centímetros} = \frac{\text{tiempo}}{58}$$

$$\text{Distancia en pulgadas} = \frac{\text{tiempo}}{148}$$

O se puede utilizar la velocidad del sonido, la cual es 340m/s

Programa de aplicación con Arduino UNO r3 y Arduino Duemilanove

Usando la velocidad del sonido.

Este programa usa la constante de la velocidad del sonido 340m/s para calcular la distancia recorrida por éste, desde que sale del sensor hasta que rebota y envía los datos por el puerto serie para ser visualizado en el monitor serial del Arduino IDE, la conexión del sensor al Arduino se muestra en la última figura.

/* Programa para modulo ultrasonido.

Usa la velocidad del sonido en el aire a temperatura ambiente 340m/s (0.034 cm/us) para completar la ecuación que define la distancia entre el sensor y un objeto capaz de rebotar la onda de sonido.

Creado: Octubre del 2013
por William Galvis.

Este código de ejemplo es de dominio público

<http://www.microkitselectronica.com/images/pdf/ultrasonido.pdf>

*/

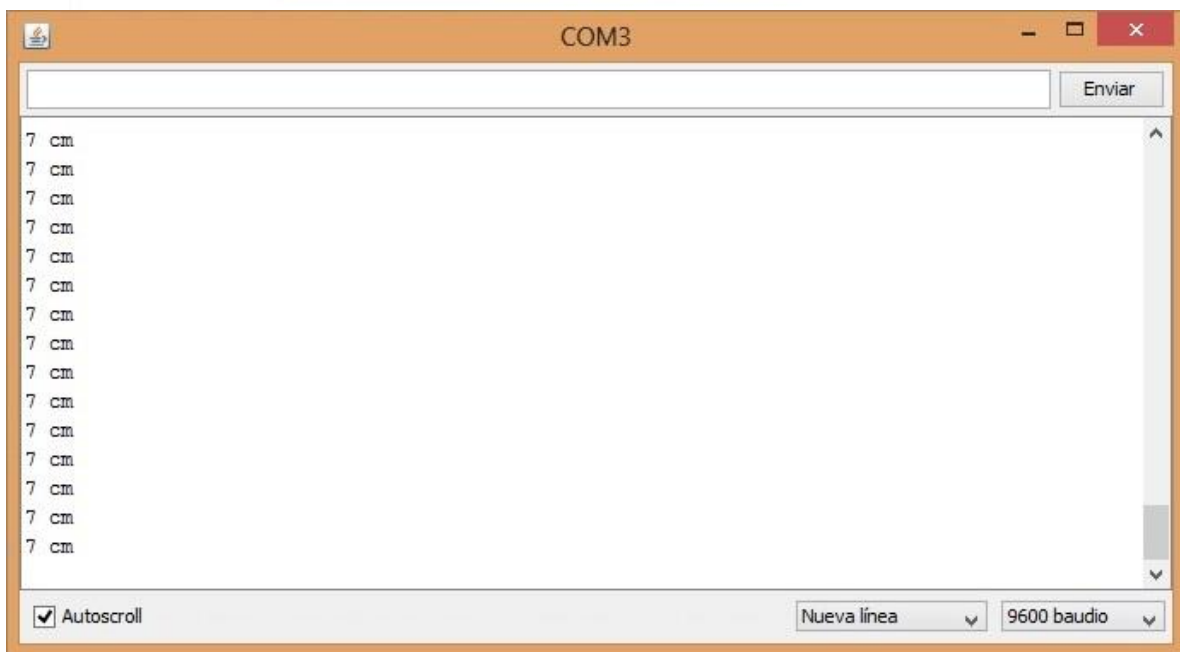
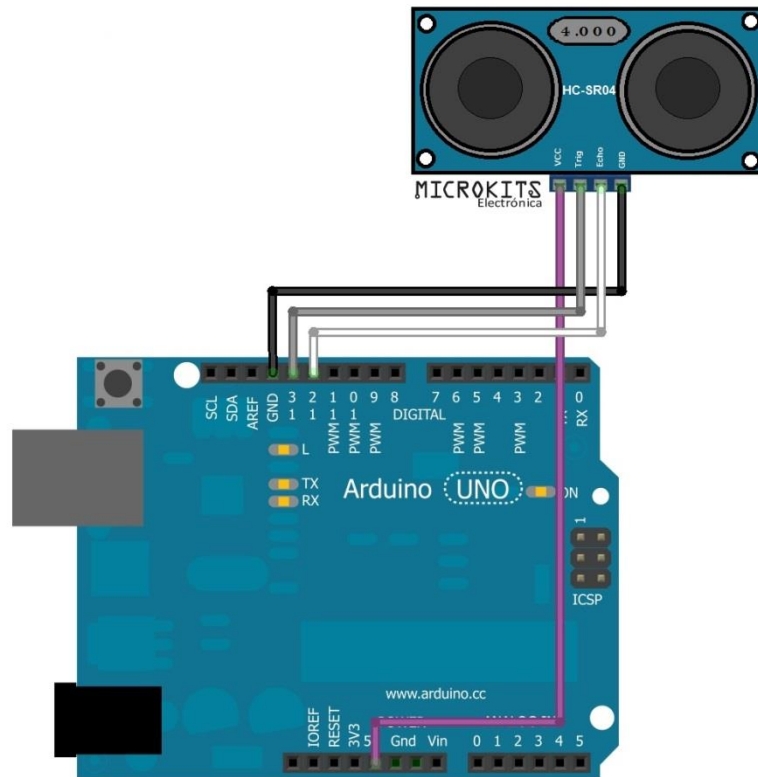
```
#define trigPin 12
#define echoPin 13
```

```
void setup() {
  Serial.begin(9600);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
}
```

```
void loop() {
  int duracion, distancia;
  digitalWrite(trigPin, HIGH); // pulso de disparo
  delayMicroseconds(1000); // dura 1000 microsegundos
  digitalWrite(trigPin, LOW); // y se apaga
  duracion=pulseIn(echoPin, HIGH); // se mide el tiempo del pin echo en alto (us)
  distancia=(duracion/2)*0.034; // distancia será la mitad del tiempo por velocidad
  Serial.print(distancia); // se envían los datos por serial
  Serial.print(" cm \n"); // se escribe la unidad de medida y un caracter de nueva línea
  delay(100); // envía datos cada 100 ms
}
```

E-Mail: info@microkitselectronica.com - www.microkitselectronica.com

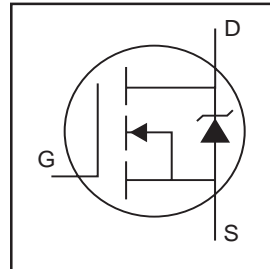
Calle: 41 No. 51 15, local 141, Teléfono: 232 77 84,
Centro Comercial Paseo Bolívar, Carabobo con los Huesos, Medellín, Antioquia, Colombia.



IRFZ44N

HEXFET® Power MOSFET

- Advanced Process Technology
- Ultra Low On-Resistance
- Dynamic dv/dt Rating
- 175°C Operating Temperature
- Fast Switching
- Fully Avalanche Rated

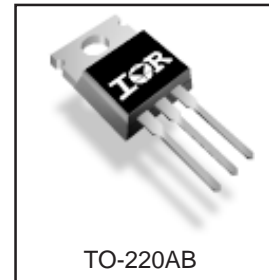


$V_{DS} = 55V$
$R_{DS(on)} = 17.5m\Omega$
$I_D = 49A$

Description

Advanced HEXFET® Power MOSFETs from International Rectifier utilize advanced processing techniques to achieve extremely low on-resistance per silicon area. This benefit, combined with the fast switching speed and ruggedized device design that HEXFET power MOSFETs are well known for, provides the designer with an extremely efficient and reliable device for use in a wide variety of applications.

The TO-220 package is universally preferred for all commercial-industrial applications at power dissipation levels to approximately 50 watts. The low thermal resistance and low package cost of the TO-220 contribute to its wide acceptance throughout the industry.



TO-220AB

Absolute Maximum Ratings

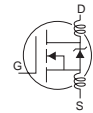
	Parameter	Max.	Units
$I_D @ T_C = 25^\circ C$	Continuous Drain Current, $V_{GS} @ 10V$	49	A
$I_D @ T_C = 100^\circ C$	Continuous Drain Current, $V_{GS} @ 10V$	35	
I_{DM}	Pulsed Drain Current ①	160	
$P_D @ T_C = 25^\circ C$	Power Dissipation	94	W
	Linear Derating Factor	0.63	W/°C
V_{GS}	Gate-to-Source Voltage	± 20	V
I_{AR}	Avalanche Current ①	25	A
E_{AR}	Repetitive Avalanche Energy ①	9.4	mJ
dv/dt	Peak Diode Recovery dv/dt ③	5.0	V/ns
T_J	Operating Junction and	-55 to + 175	°C
T_{STG}	Storage Temperature Range		
	Soldering Temperature, for 10 seconds	300 (1.6mm from case)	
	Mounting torque, 6-32 or M3 screw	10 lbf•in (1.1N•m)	

Thermal Resistance

	Parameter	Typ.	Max.	Units
$R_{\theta JC}$	Junction-to-Case	—	1.5	°C/W
$R_{\theta CS}$	Case-to-Sink, Flat, Greased Surface	0.50	—	
$R_{\theta JA}$	Junction-to-Ambient	—	62	

Electrical Characteristics @ $T_J = 25^\circ\text{C}$ (unless otherwise specified)

	Parameter	Min.	Typ.	Max.	Units	Conditions
$V_{(BR)DSS}$	Drain-to-Source Breakdown Voltage	55	—	—	V	$V_{GS} = 0V, I_D = 250\mu A$
$\Delta V_{(BR)DSS}/\Delta T_J$	Breakdown Voltage Temp. Coefficient	—	0.058	—	V/°C	Reference to $25^\circ\text{C}, I_D = 1\text{mA}$
$R_{DS(on)}$	Static Drain-to-Source On-Resistance	—	—	17.5	mΩ	$V_{GS} = 10V, I_D = 25A$ ④
$V_{GS(th)}$	Gate Threshold Voltage	2.0	—	4.0	V	$V_{DS} = V_{GS}, I_D = 250\mu A$
g_{fs}	Forward Transconductance	19	—	—	S	$V_{DS} = 25V, I_D = 25A$ ④
I_{DSS}	Drain-to-Source Leakage Current	—	—	25	μA	$V_{DS} = 55V, V_{GS} = 0V$
		—	—	250		$V_{DS} = 44V, V_{GS} = 0V, T_J = 150^\circ\text{C}$
I_{GSS}	Gate-to-Source Forward Leakage	—	—	100	nA	$V_{GS} = 20V$
	Gate-to-Source Reverse Leakage	—	—	-100		$V_{GS} = -20V$
Q_g	Total Gate Charge	—	—	63	nC	$I_D = 25A$
Q_{gs}	Gate-to-Source Charge	—	—	14		$V_{DS} = 44V$
Q_{gd}	Gate-to-Drain ("Miller") Charge	—	—	23		$V_{GS} = 10V$, See Fig. 6 and 13
$t_{d(on)}$	Turn-On Delay Time	—	12	—	ns	$V_{DD} = 28V$
t_r	Rise Time	—	60	—		$I_D = 25A$
$t_{d(off)}$	Turn-Off Delay Time	—	44	—		$R_G = 12\Omega$
t_f	Fall Time	—	45	—		$V_{GS} = 10V$, See Fig. 10 ④
L_D	Internal Drain Inductance	—	4.5	—	nH	Between lead, 6mm (0.25in.) from package and center of die contact
L_S	Internal Source Inductance	—	7.5	—		
C_{iss}	Input Capacitance	—	1470	—	pF	$V_{GS} = 0V$
C_{oss}	Output Capacitance	—	360	—		$V_{DS} = 25V$
C_{rss}	Reverse Transfer Capacitance	—	88	—		$f = 1.0\text{MHz}$, See Fig. 5
E_{AS}	Single Pulse Avalanche Energy②	—	530⑤	150⑥	mJ	$I_{AS} = 25A, L = 0.47\text{mH}$

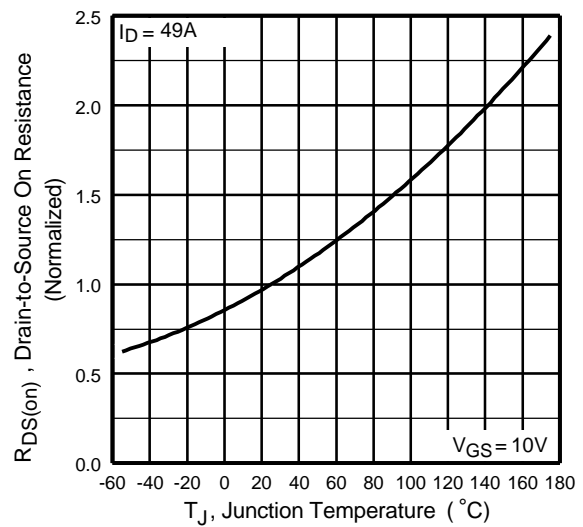
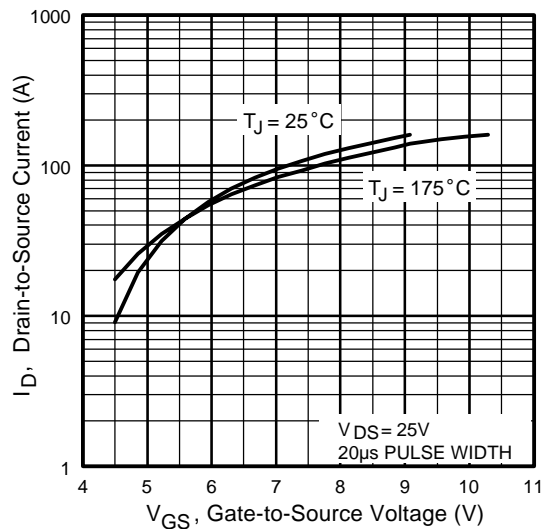
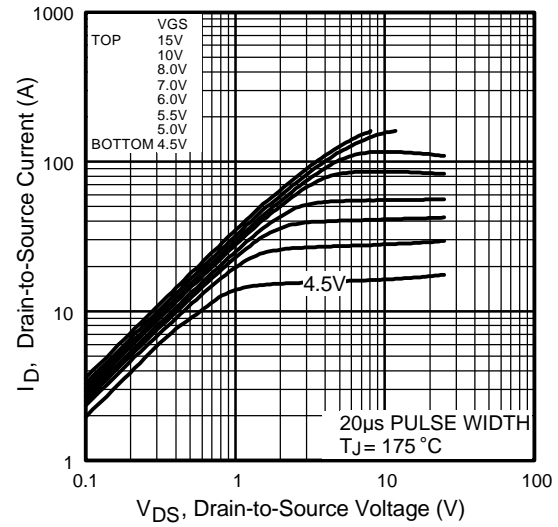
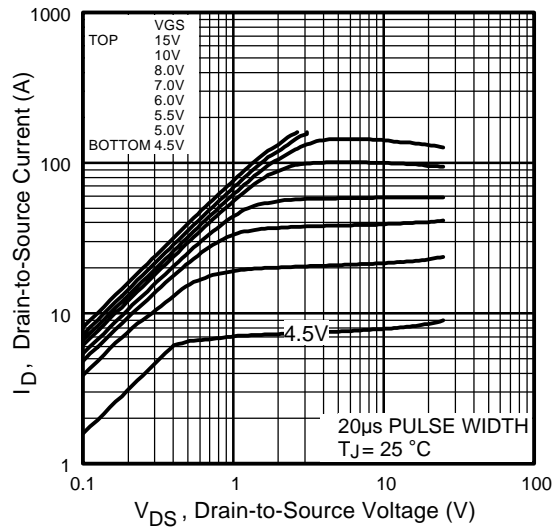


Source-Drain Ratings and Characteristics

	Parameter	Min.	Typ.	Max.	Units	Conditions
I_S	Continuous Source Current (Body Diode)	—	—	49	A	MOSFET symbol showing the integral reverse p-n junction diode.
I_{SM}	Pulsed Source Current (Body Diode)①	—	—	160		
V_{SD}	Diode Forward Voltage	—	—	1.3	V	$T_J = 25^\circ\text{C}, I_S = 25A, V_{GS} = 0V$ ④
t_{rr}	Reverse Recovery Time	—	63	95	ns	$T_J = 25^\circ\text{C}, I_F = 25A$
Q_{rr}	Reverse Recovery Charge	—	170	260	nC	$di/dt = 100A/\mu s$ ④
t_{on}	Forward Turn-On Time	Intrinsic turn-on time is negligible (turn-on is dominated by $L_S + L_D$)				

Notes:

- ① Repetitive rating; pulse width limited by max. junction temperature. (See fig. 11)
- ② Starting $T_J = 25^\circ\text{C}$, $L = 0.48\text{mH}$
 $R_G = 25\Omega$, $I_{AS} = 25A$. (See Figure 12)
- ③ $I_{SD} \leq 25A$, $di/dt \leq 230A/\mu s$, $V_{DD} \leq V_{(BR)DSS}$,
 $T_J \leq 175^\circ\text{C}$
- ④ Pulse width $\leq 400\mu s$; duty cycle $\leq 2\%$.
- ⑤ This is a typical value at device destruction and represents operation outside rated limits.
- ⑥ This is a calculated value limited to $T_J = 175^\circ\text{C}$.



IRFZ44N

International
IR Rectifier

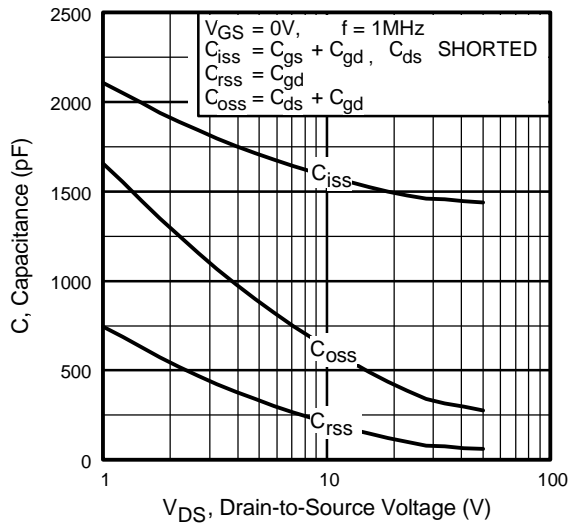


Fig 5. Typical Capacitance Vs. Drain-to-Source Voltage

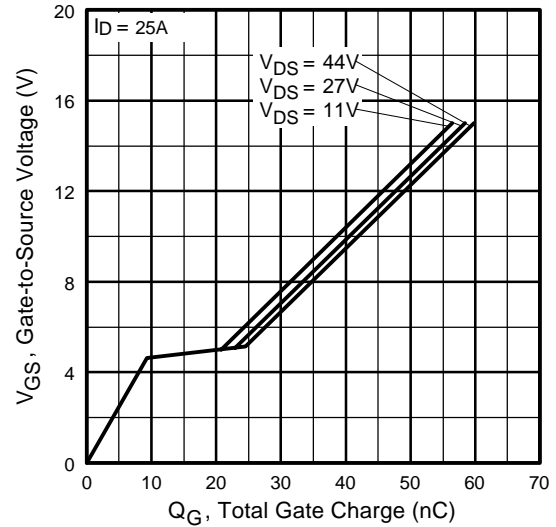


Fig 6. Typical Gate Charge Vs. Gate-to-Source Voltage

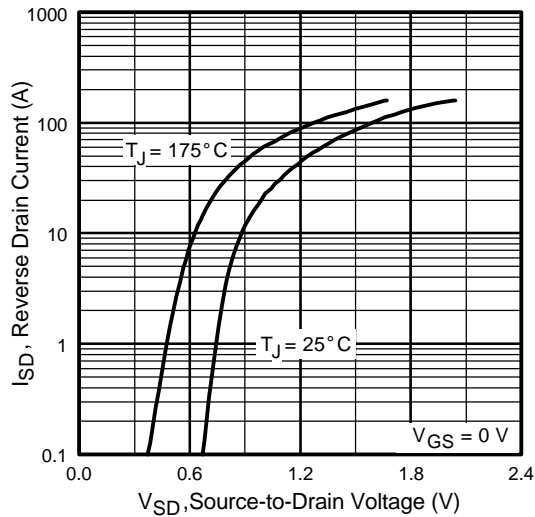


Fig 7. Typical Source-Drain Diode Forward Voltage

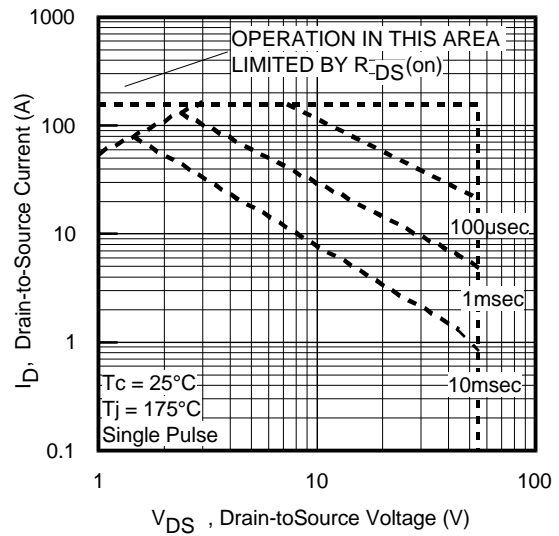


Fig 8. Maximum Safe Operating Area

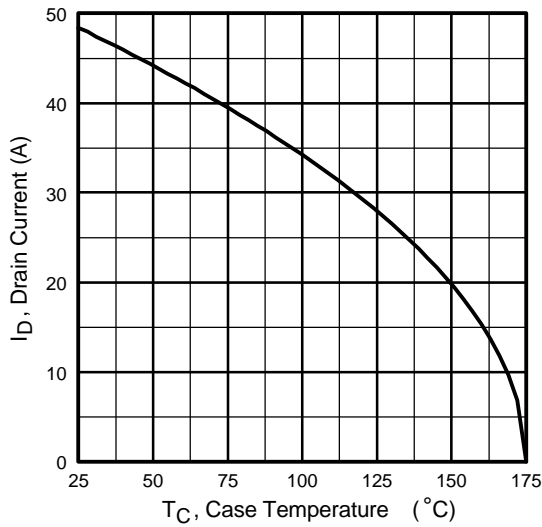


Fig 9. Maximum Drain Current Vs. Case Temperature

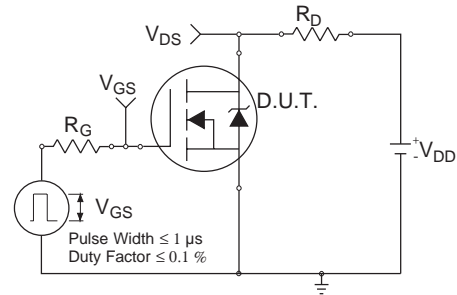


Fig 10a. Switching Time Test Circuit

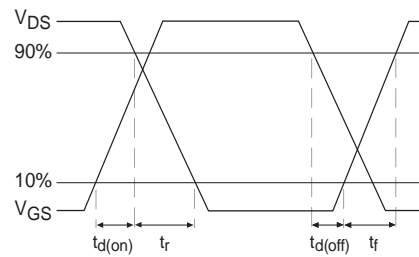


Fig 10b. Switching Time Waveforms

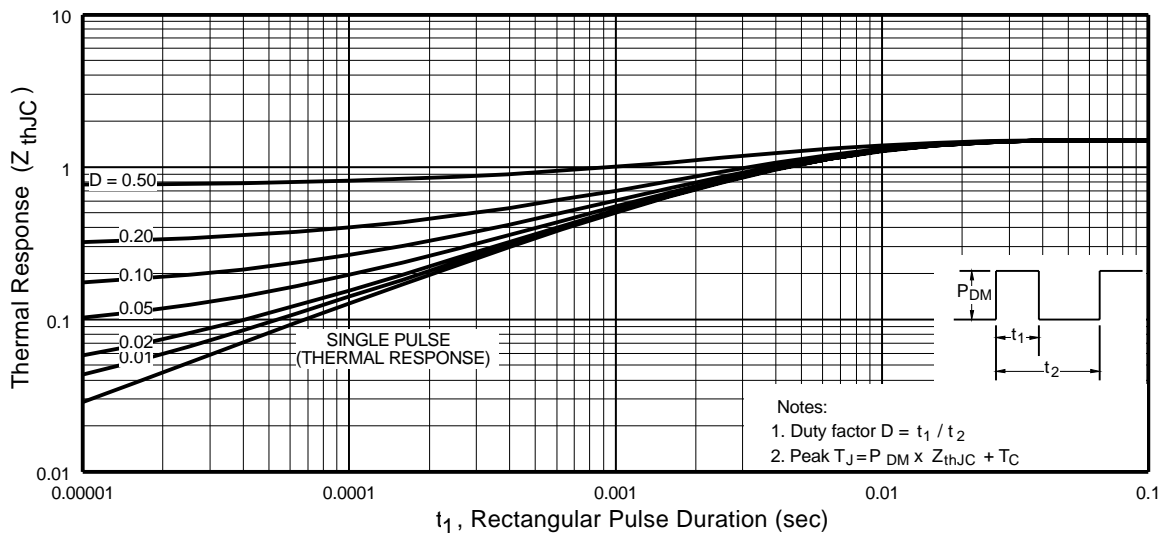


Fig 11. Maximum Effective Transient Thermal Impedance, Junction-to-Case

IRFZ44N

International
IR Rectifier

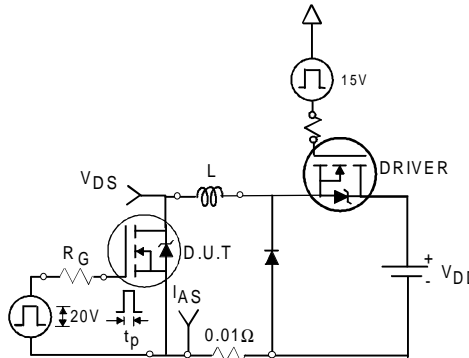


Fig 12a. Unclamped Inductive Test Circuit

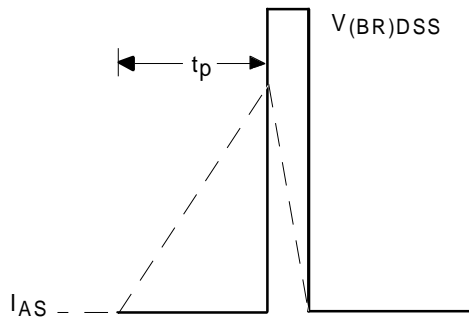


Fig 12b. Unclamped Inductive Waveforms

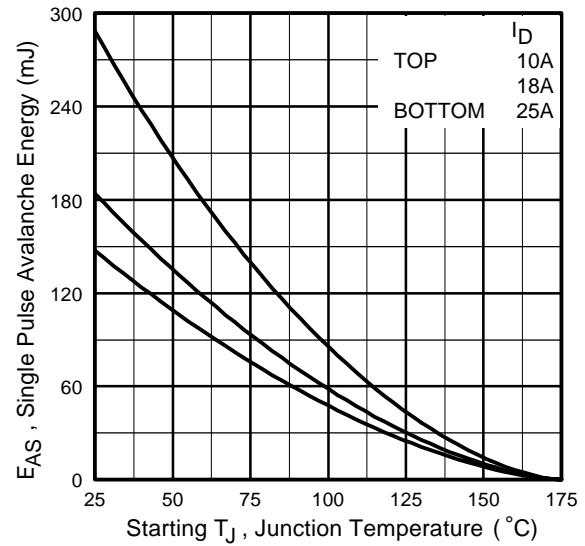


Fig 12c. Maximum Avalanche Energy Vs. Drain Current

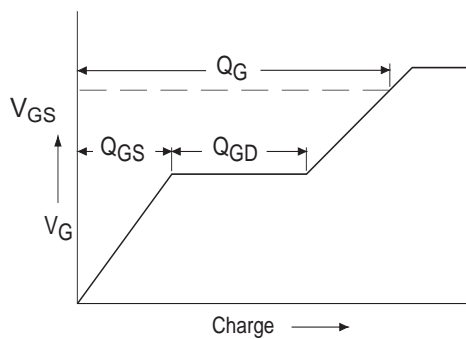


Fig 13a. Basic Gate Charge Waveform

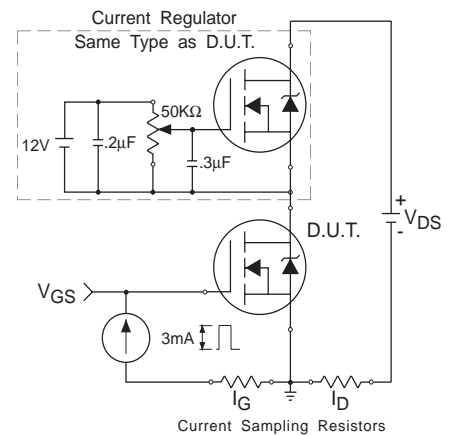
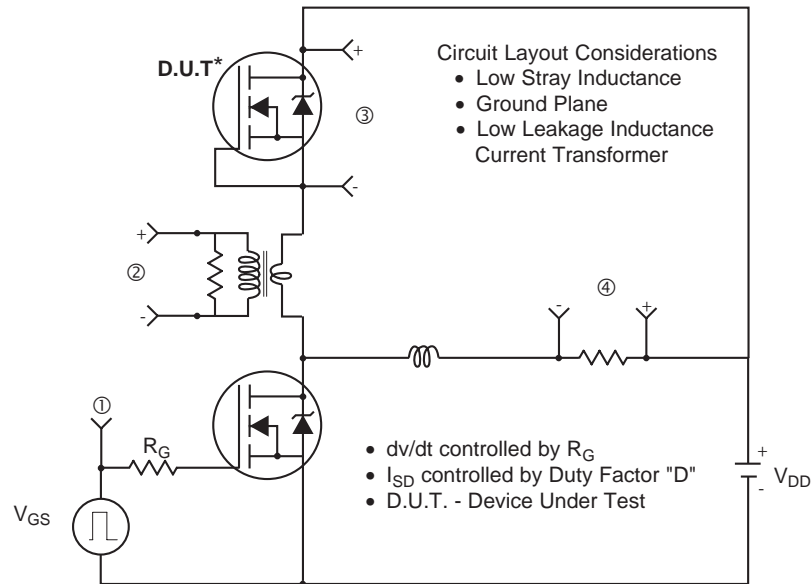
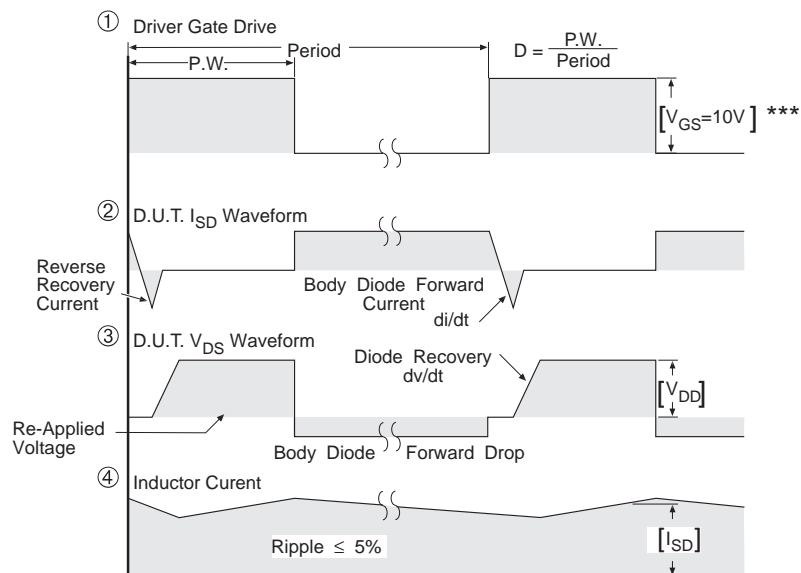


Fig 13b. Gate Charge Test Circuit

Peak Diode Recovery dv/dt Test Circuit



* Reverse Polarity of D.U.T for P-Channel



*** $V_{GS} = 5.0V$ for Logic Level and 3V Drive Devices

Fig 14. For N-channel HEXFET® power MOSFETs

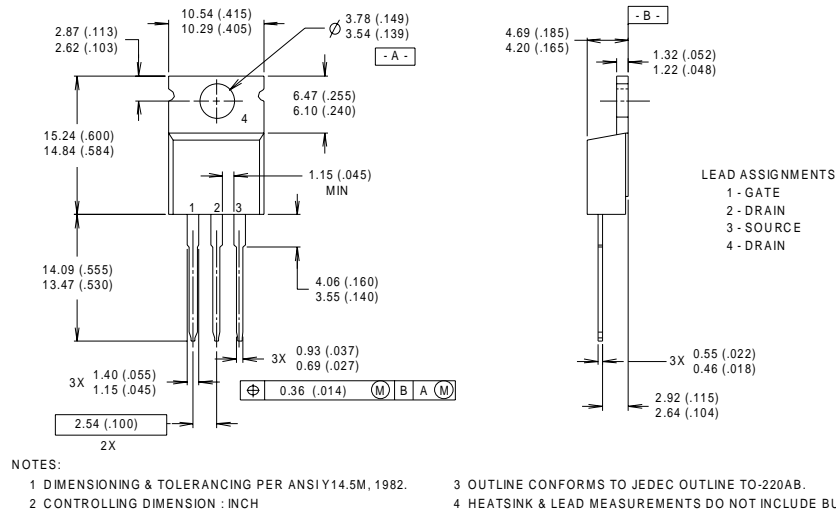
IRFZ44N

International
IR Rectifier

Package Outline

TO-220AB

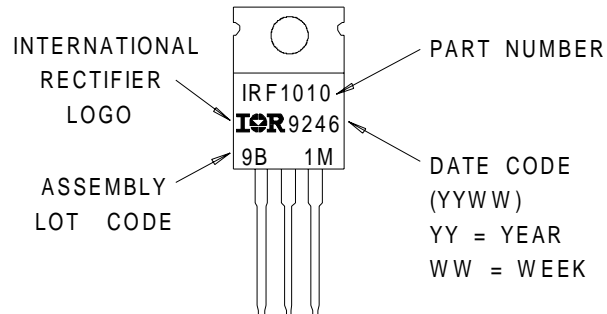
Dimensions are shown in millimeters (inches)



Part Marking Information

TO-220AB

EXAMPLE : THIS IS AN IRF1010
WITH ASSEMBLY
LOT CODE 9B1M



Data and specifications subject to change without notice.
This product has been designed and qualified for the Automotive [Q101] market.
Qualification Standards can be found on IR's Web site.

International
IR Rectifier

IR WORLD HEADQUARTERS: 233 Kansas St., El Segundo, California 90245, USA Tel: (310) 252-7105
TAC Fax: (310) 252-7903

Visit us at www.irf.com for sales contact information.01/01

www.irf.com

Note: For the most current drawings please refer to the IR website at:
<http://www.irf.com/package/>

BS LIGHT

Especificaciones Producto

Rollo led 5mts 300 leds 5050

Ref.: TF5-xx-AP-300



Rollo led 5mts 300 leds 5050

Características:

Funcionamiento a 24VDC max.

Led de alto brillo y bajo consumo BSLIGHT

Temperatura de trabajo: -20 °C~50 °C

Larga vida. Más de 50.000h

Adhesivo de doble cara marca 3M

Posibilidad de regulación

Protección IP33

Fondo de la tira en blanco o marrón

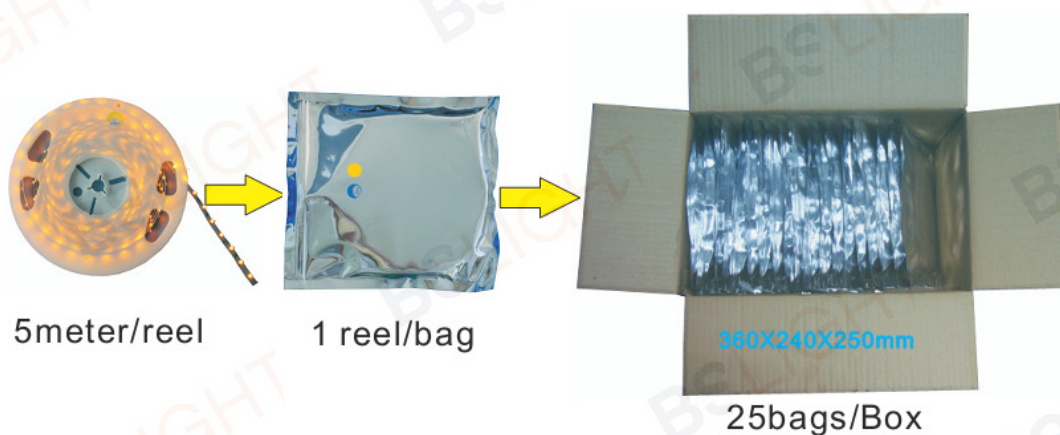


Medidas:

Top View



Section



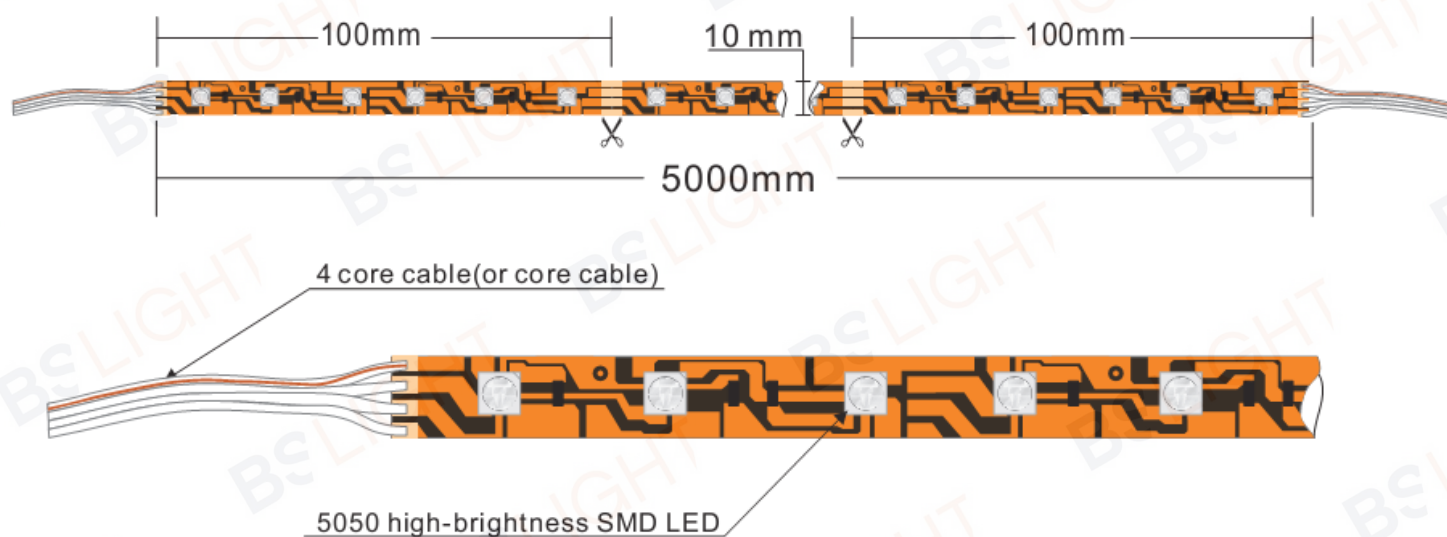
Ángulo de visión: 120degrees

Temperatura de trabajo:-20°~50°

Temperatura de almacenaje:-30°~80°

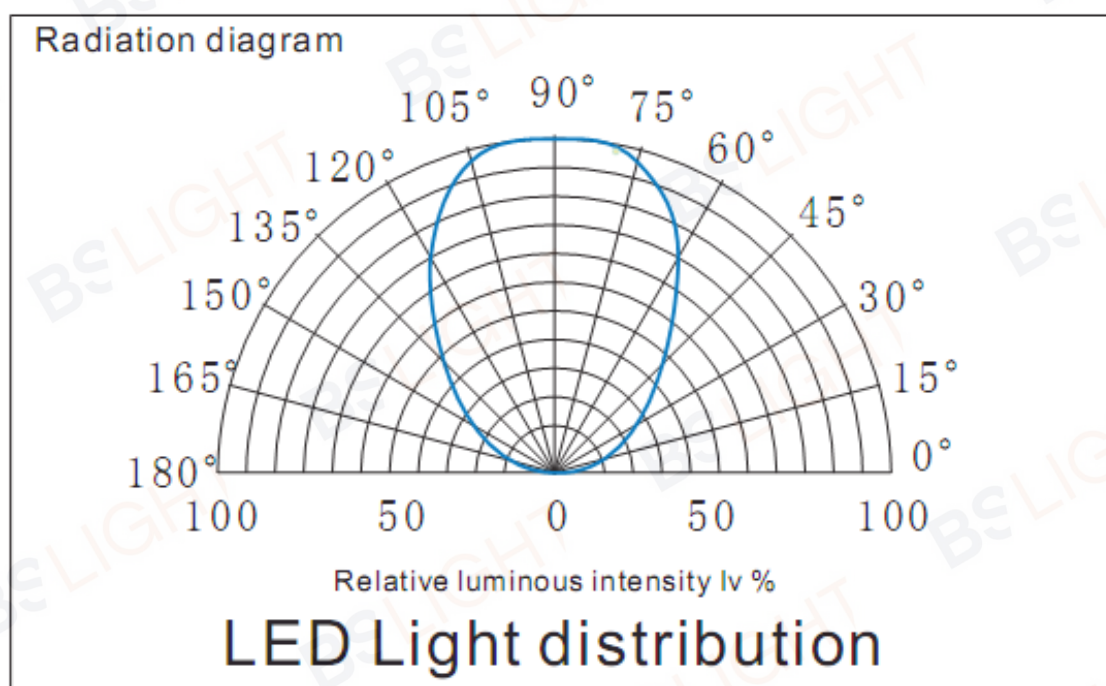
Presentación: Se entrega en rollos de 5mts en bolsa antiestática + información técnica de instalación de los equipos BSLIGHT.

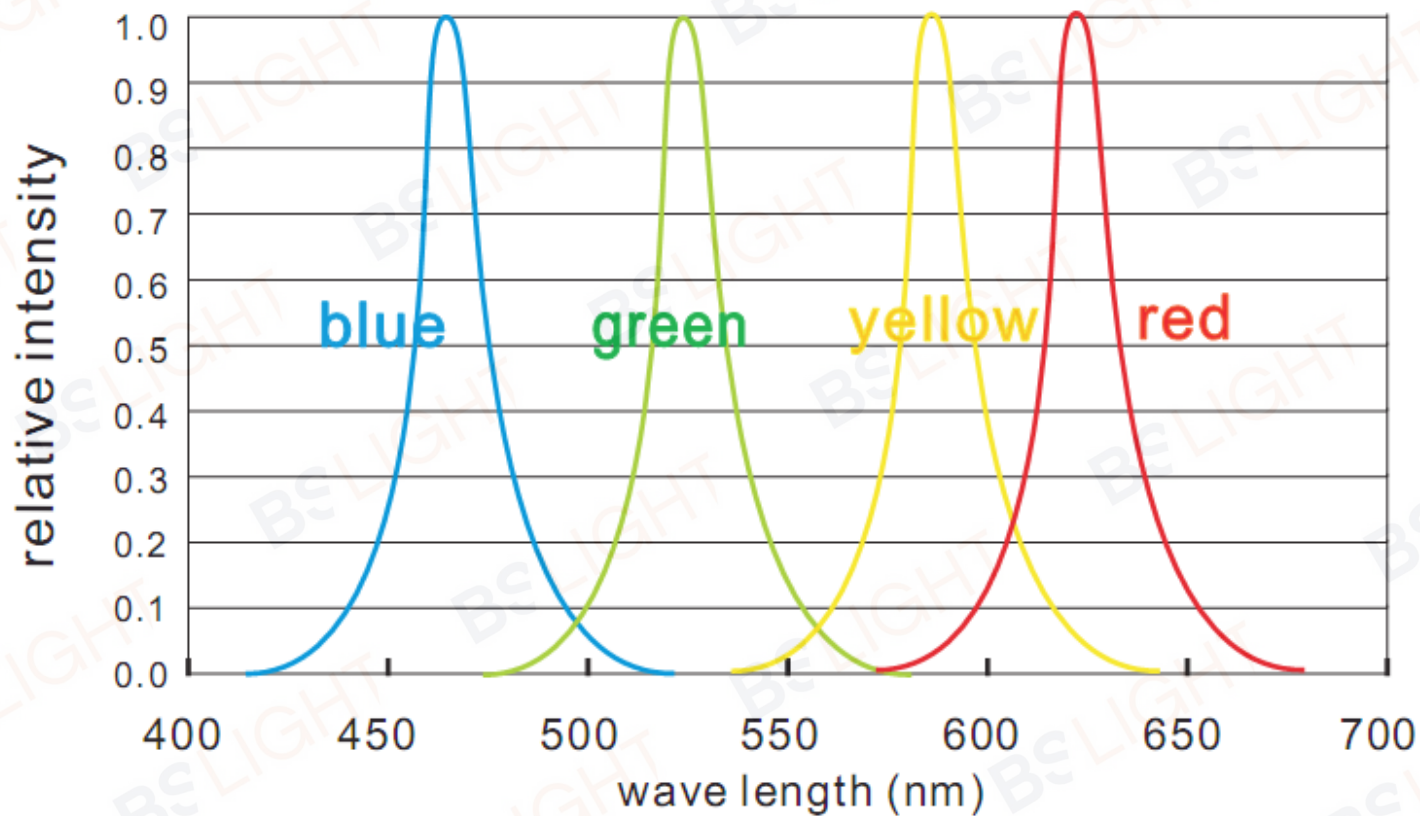
Instalar sobre superficies metálicas para su correcto funcionamiento



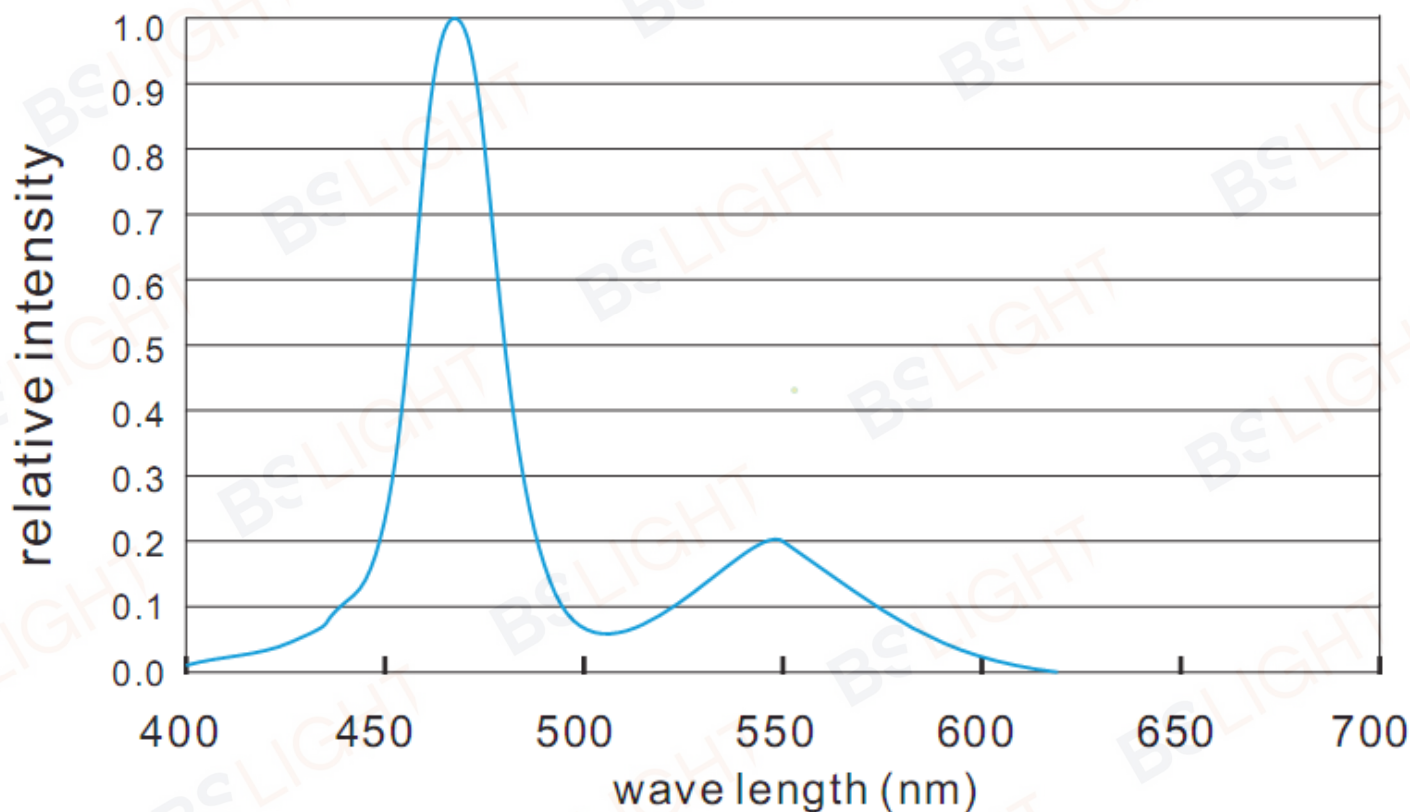
Esta tira led esta fabricada con chip 5050 de alto rendimiento

Ribbon series : 5050 60LED/meter						
Part number	Color	LED QTY/meter	Lumen/meter	Voltage	Power/meter	packing means
TF5RGB-300	Red	60	60	DC24V	14.4 Watt	5meter/reel
	Green		120			
	Blue		75			
TF5(x)-AP-300	Red	60	180	DC24V	14.4 watt	5meter/reel
TF5(x)-AP-300	Yellow	60	180	DC24V	14.4 watt	5meter/reel
TF5(x)-AP-300	Blue	60	225	DC24V	14.4 watt	5meter/reel
TF5(x)-AP-300	Green	60	480	DC24V	14.4 watt	5meter/reel
TF5(x)-AP-300	White	60	1080	DC24V	14.4 watt	5meter/reel
TF5(x)-AP-300	Warm White	60	1170	DC24V	14.4 watt	5meter/reel
TF5(x)-AP-300	Pink	60	360	DC24V	14.4 watt	5meter/reel



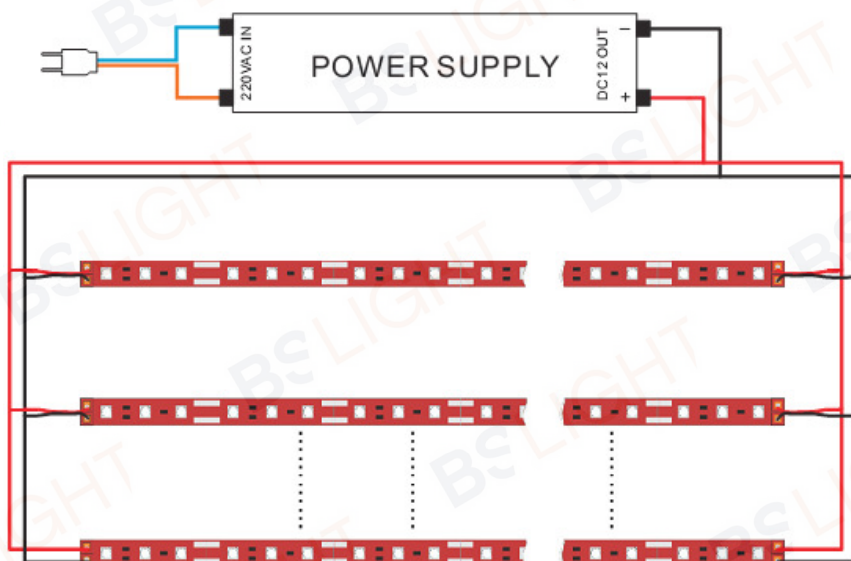


for: red, blue, green, yellow



for: white LED

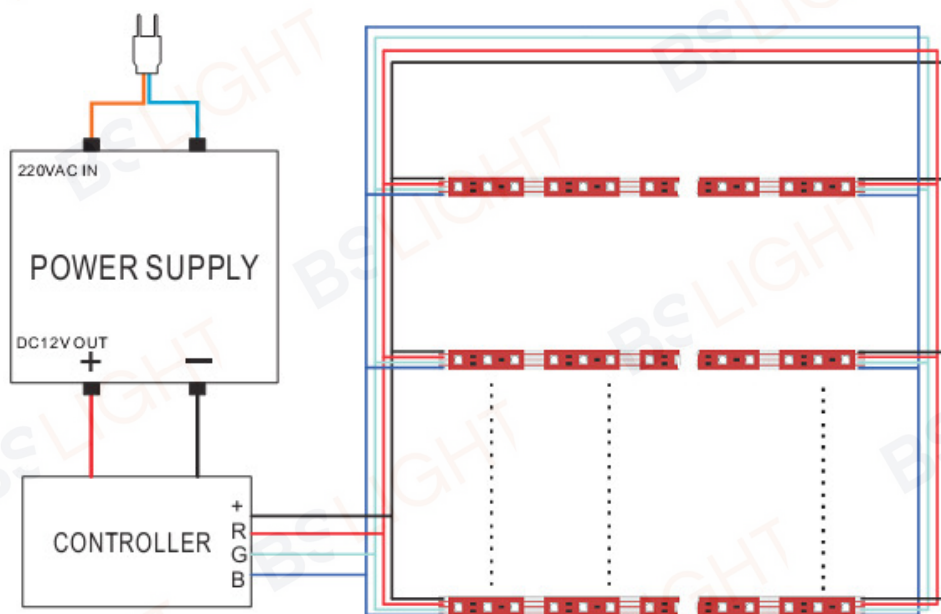
Como conectar las tiras a la fuente



Notas:

1. Máximo 5 metros en serie
2. Mejor para suministrar la energía unir el principio y final de la tira.
3. Tira flexible debe ser instalado en un espacio proporcional y uniformemente.
4. Fuente de alimentación debe ser DC 24V de la marca BSLIGHT.
5. Asegúrese que el positivo y el negativo no se invierten.

Conexión de sistemas RGB:



Notas:

1. Máximo 5 metros en serie
2. Mejor para suministrar la energía unir el principio y final de la tira.
3. Tira flexible debe ser instalado en un espacio proporcional y uniformemente.
4. Fuente de alimentación debe ser DC 24V de la marca BSLIGHT.
5. Asegúrese de lo positivo, el negativo y los cables R G B no se invierten.

Ante cualquier duda, contacte con su suministrador habitual de productos BSLight o con el fabricante BSV Electronic que le resolverán sus dudas.

Para una correcta disipación del calor producido, es imprescindible su instalación sobre perfiles de aluminio BSLIGHT o cualquier otro dispositivo de disipación alternativo.

La información técnica indicada pertenece a BSV Electronic. Queda prohibida su reimpresión o utilización sin previo aviso de su propietario BSV Electronic.